How does chunking help working memory?

Thalmann, Mirko ; Souza, Alessandra S ; Oberauer, Klaus

Abstract: Chunking is the recoding of smaller units of information into larger, familiar units. Chunking is often assumed to help bypassing the limited capacity of working memory (WM). We investigate how chunks are used in WM tasks, addressing three questions: (a) Does chunking reduce the load on WM? Across four experiments chunking benefits were found not only for recall of the chunked but also of other not-chunked information concurrently held in WM, supporting the assumption that chunking reduces load. (b) Is the chunking benefit independent of chunk size? The chunking benefit was independent of chunk size only if the chunks were composed of unique elements, so that each chunk could be replaced by its first element (Experiment 1), but not when several chunks consisted of overlapping sets of elements, disabling this replacement strategy (Experiments 2 and 3). The chunk-size effect is not due to differences in rehearsal duration as it persisted when participants were required to perform articulatory suppression (Experiment 3). Hence, WM capacity is not limited to a fixed number of chunks regardless of their size. (c) Does the chunking benefit depend on the serial position of the chunk? Chunks in early list positions improved recall of other, not-chunked material, but chunks at the end of the list did not. We conclude that a chunk reduces the load on WM via retrieval of a compact chunk representation from long-term memory that replaces the representations of individual elements of the chunk. This frees up capacity for subsequently encoded material.

DOI: https://doi.org/10.1037/xlm0000578

Research Article

How Does Chunking Help Working Memory?

Mirko Thalmann, Alessandra S. Souza, & Klaus Oberauer

University of Zurich, Switzerland

Thalmann, M., Souza, A. S., & Oberauer, K. (2019). How does chunking help working memory? *Journal of Experimental Psychology: Learning, Memory and Cognition, 45*, 37-55. doi:10.1037/xlm0000578

Author Note

Mirko Thalmann, Alessandra S. Souza, and Klaus Oberauer, Department of Psychology, University of Zurich, Switzerland. This research was supported by a grant from the Swiss National Science Foundation to K. Oberauer (project 149193). Correspondence should be addressed to Mirko Thalmann, Department of Psychology, Cognitive Psychology Unit, University of Zürich, Binzmühlestrasse 14/22, 8050 Zurich, Switzerland. E-mail: mirkothalmann@hotmail.com

Abstract

Chunking is the recoding of smaller units of information into larger, familiar units. Chunking is often assumed to help bypassing the limited capacity of working memory (WM). We investigate how chunks are used in WM tasks, addressing three questions: (1) Does chunking reduce the load on WM? Across four experiments chunking benefits were found not only for recall of the chunked but also of other not-chunked information concurrently held in WM, supporting the assumption that chunking reduces load. (2) Is the chunking benefit independent of chunk size? The chunking benefit was independent of chunk size only if the chunks were composed of unique elements, so that each chunk could be replaced by its first element (Experiment 1), but not when several chunks consisted of overlapping sets of elements, disabling this replacement strategy (Experiments 2 and 3). The chunk-size effect is not due to differences in rehearsal duration as it persisted when participants were required to perform articulatory suppression (Experiment 3). Hence, WM capacity is not limited to a fixed number of chunks regardless of their size. (3) Does the chunking benefit depend on the serial position of the chunk? Chunks in early list positions improved recall of other, not-chunked material, but chunks at the end of the list did not. We conclude that a chunk reduces the load on WM via retrieval of a compact chunk representation from long-term memory that replaces the representations of individual elements of the chunk. This frees up capacity for subsequently encoded material.

The scripts for all the experimental procedures and the raw data of the present article are available on the following Open-Science Framework webpage: https://osf.io/jjfbh/?view_only=3ebcbef89c3545019f6fde0fe28729f3.

Keywords: Short-Term Memory, Working Memory, Long-Term Memory, Chunk

When people are required to remember a number of items over a brief retention interval they usually cannot recall more than a few of them (Brener, 1940). For example, when presented with a random sequence of geometric figures, participants can only recall about five or six of them in correct order. This limitation in remembering information over the short term is due to a limited-capacity store called working memory (WM). Although WM is severely limited in its capacity, some people are able to remember lists far exceeding this capacity. For example, when chess players at beginner level are briefly presented with middle-game positions in a chess game, they can recall about four pieces in the correct location (Chase & Simon, 1973). In contrast, players at advanced level or at master level can recall about eight or 16 pieces correctly, respectively. If WM is capacity limited, how can there be such a large difference in memory for chess positions as a function of chess expertise? Chase and Simon assumed that experienced chess players encode the positions as larger perceptual chunks, "each consisting of a familiar sub configuration of pieces" (p. 80). This interpretation is supported by the fact that, when presented with random constellations of chess pieces (i.e., shuffling the pieces of middle games), the number of pieces remembered by master players dropped to the level of beginners. It follows that pre-existing knowledge in terms of chunks can boost immediate memory. The present study focuses on this increase in immediate memory and asks in more detail what processes contribute to it.

The process of chunking was first described by Miller (1956) as the recoding of several presented stimuli into a single familiar unit or chunk. Miller proposed that chunking is achieved by grouping or organizing a sequence of inputs, and recoding it with a concise name. Therefore, remembering just the name essentially reduces the storage load on WM, arguably freeing capacity for storage of additional information. The reference to "familiar" units can be understood here as referring to the reliance on long-term memory (LTM)

representations. In a similar vein, Cowan (2001) defines chunks as groups of items that have strong, pre-existing associations to each other but weak associations to other items. To summarize, both authors highlight the importance of LTM in their definition of a chunk. The present work sought to make a step towards a deeper understanding of how people make use of chunks in WM tasks. In particular, we were interested in the question how chunks can reduce the load on WM.

Following Miller, several theorists have assumed that WM capacity is limited in terms of the number of chunks. Most prominently, Cowan (2001) proposed that about four chunks are available at a certain point in time in the focus of attention. Other researchers have also embraced the chunking idea, although they have different views on the exact number of chunks that can be held in WM (Chase & Simon, 1973; Gobet & Clarkson, 2004).

Evidence for the fixed-chunk hypothesis comes from experiments that varied chunk size and observed an approximately equal number of chunks recalled across different chunk sizes (Chen & Cowan, 2005, 2009; Cowan, Chen, & Rouder, 2004). In Chen and Cowan (2005), for example, participants were first trained to remember the pairings in a set of word pairs. Training also involved a set of single words for which participants had to remember that they were not paired with another word. Training proceeded until recall was 100% accurate. At this point, each individual trained word (henceforth singleton) and each of the word pairs were considered a chunk. Next, participants attempted immediate serial recall of word lists ranging from 4 to 12 words. The lists consisted either of learned word pairs, learned singletons, or new words. When recall of words was scored regardless of order, participants recalled approximately 3 chunks across all conditions (i.e., twice as many words from lists of word pairs than from lists of singletons or new words), consistent with the assumption that WM capacity is limited to a fixed number of chunks. Chen and Cowan

(2009) showed that the constant number of recalled chunks can be observed best when participants had to engage in concurrent articulation ("articulatory suppression", AS) during encoding. The authors concluded that the fixed capacity limit on chunks is most directly reflected in performance when articulatory rehearsal of phonological representations is prevented.

These experiments leave open two possibilities of how chunks help immediate recall. First, chunks require less capacity and therefore free up capacity in WM. This would be the case, for example, if remembering a single word required the same capacity as remembering a learned pair, because all were remembered as one chunk (Cowan, 2001). This account posits that a chunk is represented in WM independently of its composite elements, for example by a concise name as suggested by Miller. In contrast, a second possibility is that information from LTM assists in the reconstruction of the complete chunk from partial information in WM (Hulme, Maughan, & Brown, 1991; Hulme & Roodenrys, 1995). This account assumes that learned pairs (chunks) are maintained in WM in the same way as random pairs. However, at recall there is more LTM knowledge available for a previously learned pair compared to a random pair to assist reconstruction of the original set of elements. To distinguish these two possibilities, one needs to test memory for other items in the presence of a chunk: If chunking reduces the load on WM capacity, the presence of a chunk in a memory list should improve memory for other items maintained concurrently. In contrast, if chunks benefit only from being more successfully reconstructed at retrieval, other information in WM should not inherit that benefit.

To the best of our knowledge, only the study by (Portrat, Guida, Phénix, & Lemaire, 2016) presented mixed lists consisting of chunks and not-chunked items. However, they did

not assess whether chunks improved the retention of not-chunked items. Therefore, it is still an open question whether chunks actually reduce the load on WM.

The experiments showing evidence that WM capacity is limited by a fixed number of chunks (Chen & Cowan, 2005, 2009; Cowan et al., 2004) had an additional feature, which makes their interpretation difficult: Each pair to be remembered consisted of unique elements. That is, each word could only occur as a single word or as a member of one pair. This is not generally the case with chunks. Consider the often cited examples FBI and CIA, or any other acronym – they consist of letters that also occur in many other acronyms, and when known acronyms are included in lists of letters presented for short-term retention, each letter that figures as an element of an acronym can also occur as a singleton in the list (Portrat et al., 2016).

The use of chunks with unique elements in the experiments of Chen and Cowan has two important consequences. First, chunks can be detected instantaneously after presentation of the first word. This makes the encoding of the second word unnecessary, because participants already know the second word. Second, participants only have to remember the first word of a pair to remember both words. Even though doing so reduces the load on WM for this special kind of chunks, it is unclear whether this holds for chunks in general. Together, these two consequences of the unique-element chunks in the Chen and Cowan studies are sufficient to explain why participants remembered a fixed number of chunks: If for each learned pair they only encoded the first word into WM, then they encoded the same number of words into WM in all experimental conditions. This would be so regardless of whether or not WM capacity is limited to a fixed number of chunks. To circumvent this problem, in the current investigation we also investigated the chunking benefit with chunks consisting of not-unique elements.

The Present Study

The aims of the present study were three-fold. Our first goal was to assess whether chunking information in WM frees capacity to hold other, not-chunked information. To test for this possibility one needs to assess the impact of the chunk on the recall of other not-chunked information in WM. This prediction is best explained by means of an example. Assume that two lists have to be remembered: List 1 = F-B-I-D-Q-B, and List 2 = I-F-B-D-Q-B. In List 1, the first three items form a single chunk (i.e., F-B-I). If encoding these three letters as a chunk reduces the load on WM from 6 items to 4 items, then there will be more free capacity to hold the second half of the list (i.e., D-Q-B) in List 1 than in List 2. Consequently, short-term retention of the second half of the list should be better in List 1 than in List 2.

The second aim of the present study was to provide yet a stronger test of the tenet of Cowan (2001) that the chunking benefit is independent of chunk size because chunks are assumed to be the basic storage units in WM. To do so, we compared recall of not-chunked lists while participants had to also hold in mind a chunk varying in size (e.g., 2-item vs. 4-item chunks). If capacity is independent of chunk size, then the chunking benefit for the not-chunked lists should be of similar magnitude when participants hold a smaller or larger chunk in WM.

Third, we examined for the first time how the chunking benefit is moderated by the requirements of detecting the chunk while at the same time holding other information in WM. This point is particularly important in relation to the type of material (i.e., chunks of unique vs. not-unique elements). When chunks consist of not-unique elements, participants need to encode all individual items before they can detect a chunk. This implies that at least temporarily their WM is loaded with the individual elements of the chunk, before these elements can be replaced by a single representation of the chunk. The temporary encoding

of multiple elements could already damage other information in WM (i.e., through interference, or competition for rehearsal) before the WM load is reduced through chunking. Moreover, the reduction of WM load requires removing the individual elements from WM, so that the chunking benefit also depends on the efficiency of this process. Therefore, the chunking benefit for not-chunked information in WM might be much reduced, or even eliminated, when chunks consist of not-unique elements.

If the elements of a chunk must initially be encoded into WM individually before they are replaced by a more compact chunk, then we should also observe that the chunking benefit varies as a function of when a chunk is presented within a sequentially presented memory set. When presented at the beginning of the set, there is no other information in WM that could suffer from the temporary presence of the individual elements in WM, and these elements can be removed efficiently through a complete wipe-out of WM before encoding the compact chunk (Ecker, Oberauer, & Lewandowsky, 2014). When the chunk appears later in a trial, individual items start to interfere with earlier encoded information before they can be replaced by the chunk representation, and that replacement involves targeted removal of only the items that belong to the chunk. The selective removal of individual elements from a memory set in WM is a more difficult process than the wholesale removal of the entire set (Ecker et al., 2014). As a consequence, the chunking benefit for simultaneously maintained not-chunked information should be smaller the later a chunk is presented as part of a memory set.

We tested the predictions detailed above with four experiments. In Experiment 1, we trained participants to recall chunked lists consisting of 2 or 4 words. Each chunk consisted of unique words. Next, we asked participants to hold two lists of 2 or 4 words for an immediate serial recall test. Each list was either a randomly arranged set of words (new lists)

or a chunked list. Our main interest was in assessing whether recall of new lists was better in the presence of chunked lists (hereafter referred to as a chunking benefit) and whether this benefit was independent of chunk size. In Experiment 2, we tested the chunking benefit in conditions in which the status of the list (chunk or new) was unknown until all items of the list were presented, because the same item could occur in a chunk and in a not-chunked list. Moreover, we tested whether the load on WM from a chunked list was similar to that from holding a single-item representation (singleton) in WM. We also tested that question with AS (Experiment 3) to see whether any difference between large and small chunks could be attributed to differences in the duration of articulatory rehearsal. Finally, Experiments 2-4 investigated whether the chunking benefit depends on the serial position of the chunk within a trial.

Experiment 1

In Experiment 1, we tested the prediction that including a chunk in a memory set reduces the load on WM, thereby improving recall of other, not-chunked information maintained simultaneously with the chunk. To test this prediction, participants were presented with a memory set consisting of two short lists, followed by serial recall tests of each list. Each list was either a chunk or a new list composed of singleton words. Chunked lists were learned by heart in a training phase. As in the studies by Chen and Cowan (2005, 2009), each word could only occur either in a chunked list or a new list. If chunking frees WM capacity, then not only recall of the chunked list but also recall of the other not-chunked list on that trial should be better, compared to the condition in which both lists were new. Furthermore, we tested whether the chunk benefit was independent of chunk size. To this end we varied the length of the two lists independently of each other – new and chunked lists could comprise 2 or 4 items. If WM capacity is constrained by the number of chunks, but

not by the size of each chunk, the benefit yielded by the chunk should be of similar magnitude irrespectively of the size of the chunked list.

Method

Participants

Twenty university students (15 women; M ≈ 25 years old) took part in two 1-hour sessions. They were compensated for participation with 30 Swiss Francs or partial course credit. All participants of the experiments reported in the present paper were native speakers of German. They were required to read and sign an informed consent form before the experiment started. In the end of the experiment, they were debriefed in detail about the purpose of the study.

Materials and Procedure

All experiments were programmed and run with the Psychophysics Toolbox 3 (Brainard, 1997; Pelli, 1997) in MATLAB. Participants sat at a distance of approximately 50 cm from the computer screen (viewing distance unconstrained). They were tested in individual cabins.

We constructed a pool of 338 one- and two syllabic nouns. None of the nouns started with the same three letters in the same order. For every participant, 24 nouns were randomly selected from this pool. Half of the nouns were used as singletons to construct new lists from. The other 12 nouns were used to construct four chunked lists: two chunked lists with 2 nouns; and 2 chunked lists with four nouns. Together, the singletons and the chunked lists were used as 16 sets to be learned in a training phase at the beginning of the experiment.

Training phase. In every training cycle, the elements of the 16 sets were displayed one by one across a row of black frames in the upper part of the screen. Set presentation was self-paced: participants started the presentation of each set in a cycle by pressing the spacebar. Words in sets of one (singletons), two, or four (chunked lists) were presented from left to right, each for 1000 ms. Order of presentation of the 16 sets was randomized in every cycle. After presentation of the 16 sets, cued recall tests of all sets followed. On every test, the first word of a set (randomly selected from all sets without replacement) was presented as a probe for 1000 ms in the top half of the screen, followed by presentation of four response boxes in the bottom half. Presentation of the probe prompted participants to type all words belonging to that set, beginning with the first word (i.e., the word presented as probe). Participants could use the backspace key to correct any typing errors. They confirmed an answer by pressing the enter key. In case the tested set was a singleton or a 2-item chunked list, the remaining recall possibilities had to be skipped using the enter key. An answer was counted as correct when the first three letters of the word matched the word at that position in the set. Upper and lower case did not matter for responses to be counted as correct.

Participants completed a minimum of eight training cycles. Further training cycles were added until recall of all sets was 100% correct. Next, participants completed a distraction task. In the distraction task, participants had to judge the accuracy of 40 multiplication equations consisting of two factors in the range of 3-9. About half of the equations were correct. Participants pressed the left or right arrow key to indicate whether the displayed result of the multiplication was correct or incorrect, respectively. After the distraction task, we tested memory for the sets again via probed recall to ascertain that the chunks had been learned in LTM. As long as recall was not perfect, additional cycles

consisting of set presentation, probed recall, distraction, and again probed recall to test LTM were added. Only when all 16 word sets were recalled correctly in the LTM test, participants proceeded to the main experimental phase.

Main experimental phase. Every trial started with the presentation of a fixation cross in the middle of the screen for 500 ms. Then, two lists were presented sequentially, one in the upper and one in the lower part of the screen. Nouns within each list were presented sequentially for 1000 ms across a row of black frames. Presentation of the last noun of the first list was followed immediately by presentation of the first noun of the second list without any free time in between. The two lists were probed in random order, 500 ms after list presentation. Recall within a list proceeded in forward order. Participants typed the words using the keyboard. The same scoring as in the Training phase was applied.

We independently varied the size (2 or 4 items) and type (new or chunk) of list 1 and list 2, as well as their order of recall, resulting in 32 conditions. We aimed at replicating each condition eight times, and presenting them in random order across trials. However, due to a programming error, the 32 conditions were replicated 128 times resulting in 4096 trials that were randomized. The first 256 trials of this set were presented to the participants, resulting in an unbalanced design. In our analysis, we collapsed data across order of list presentation (list 1 and list 2) to increase the average number of data points per design cell available to be analyzed per participant (M = 16.00, SD = 3.76).

Results

Data Analysis. Frequentist statistics that are commonly used in psychological research have several shortcomings. For example, p-values express how likely the data are, given that the null hypothesis is true. However, researchers are usually interested in the reverse direction of inference: How likely is a hypothesis given the data? To overcome this

shortcoming, and some other shortcomings associated with frequentist statistics (Wagenmakers, 2007) we used Bayesian statistics for all analyses reported in the present article.

In Bayesian statistics, the believability of model parameter values – such as the effect size of an experimental manipulation – is expressed as prior distributions (hereafter priors). The priors are updated with the likelihood of the data to yield posterior distributions of the parameters. Therefore, inference based on posteriors combines all the available information about the model parameters (Kruschke, 2011). The 95% highest-density interval (95% HDI) of the posterior covers the range of parameter values that are 95% credible after the data have been observed. Hence, the 95% HDI can be used to inform about uncertainty of parameters in question. In the present work, we do not display classical confidence intervals but use HDIs of the posteriors instead, because they can be interpreted straightforwardly, which is not the case for confidence intervals (Morey, Hoekstra, Rouder, Lee, & Wagenmakers, 2016). For descriptive purposes, we also plot the standard error of the mean to show the variability of the data without the assumptions of any model.

In Bayesian statistics, models can be compared using the Bayes Factor (BF). The BF quantifies the strength of evidence in favor of one model in comparison to another, competing model, given the data. For example, we can compare a model including an effect in an ANOVA (e.g., a main effect, a two-way interaction) to a model omitting this effect. A BF in favor of the former model reflects the evidence in favor of the effect; the inverse of the BF states the evidence in favor of the null hypothesis that the effect is absent.

As a rough guideline to interpret the quantity of BFs, (Kass & Raftery, 1995) suggest that BFs between 1 and 3.2 are not worth more than a bare mention, BFs between 3.2 and 10 are substantial evidence, BFs between 10 and 100 represent strong evidence, and BFs > 100 are seen as decisive. In the present article, we computed the BF for t-tests with the BayesFactor package (Morey & Rouder, 2014) using the default priors. All other BFs were computed with self-constructed JAGS (Plummer, 2003) models using the Savage-Dickey density ratio, which provides the BF for nested models (Lee & Wagenmakers, 2014; Wetzels, Raaijmakers, Jakab, & Wagenmakers, 2009). When comparing a model in which the parameter of interest is allowed to vary freely to a model in which the parameter of interest is fixed to the value of the Null model (usually zero), the BF can be obtained by dividing the height of the prior by the height of the posterior at the parameter value of the Null model.

Whenever we used self-constructed JAGS models, we applied a two-stage procedure to determine the BFs of the effects of interest. First, we selected the best-fitting model out of a set of three models according to the DIC[1], which can be used for model selection of hierarchical Bayesian models (Spiegelhalter, Best, Carlin, & Linde, 2002). The three fitted models varied in their hierarchical structure. Model 1 assumed a random intercept for each subject but only fixed effects on all other parameters. Model 2 assumed random effects on all parameters in the model except for the highest-order interaction. Model 3 assumed random effects on all main effect parameters, but no random effects on interactions. Second, we computed the BFs for all effects of interest within the winning model with the Savage-Dickey density ratio. In the following, we only report the BFs of the winning model.

[1] The DIC relates to other information criteria as it uses a classical estimate of fit and penalizes for the effective number of parameters in a model.

Interested readers are referred to the supplementary material if they want to see which model won the DIC comparison in each case.

Serial Recall of New Lists. Our main interest was in how the accuracy of recall of new lists is influenced by the presence of a chunk, and by the size of that chunk, compared to trials in which both lists were new. If storing a chunk frees WM capacity, we should observe a main effect of list type, with better recall of new lists when the other list is a chunk than when it is another new list. If chunks impose a constant load on WM regardless of their size, then recall of a new list should not vary as function of chunk size, whereas recall would decrease when the other list is a new list with size 4 compared to size 2, resulting in an interaction between other-list type (new or chunk) and other-list size (2 or 4 items). The relevant data are displayed in Figure 1 and in Figure 2. Note that recall of new lists with 2 and 4 items are presented in different subpanels in Figure 1.

We analyzed serial recall accuracy of new lists with two Bayesian linear regressions (i.e., one for each tested set size[2]). We entered the effects for type of the other list (new list or chunk), size of the other list (2 or 4 items), time of recall of the new list to be analyzed (first vs. second), and all higher-order interactions between these predictors.

[2] The reason for running separate analyses is that we wanted to obtain the evidence in favor of the disordinal Size Other x Type Other interaction (left panel in Figure 1). Evidence for a disordinal or cross-over interaction is more convincing than for an ordinal interaction because a disordinal interaction remains when a non-linear transformation is applied to the dependent variable (Loftus, 1978).

The statistical evidence for the fixed effects in the two separate regression models is shown in Table 1. There was decisive evidence in favor of the main effect of other-list type, which means that a new list was remembered better when it was presented together with a chunk compared to another new list. This finding supports the hypothesis that chunks reduce the load on WM. The evidence was also decisive in favor of the interaction between other-list type and other-list size. This indicates that increasing the size of a chunk leads to a smaller additional load on WM than increasing the size of a new list. Whether there is any effect of chunk size at all was evaluated in an additional analysis reported below.

The other reported main effects show that new lists are remembered worse in the presence of long than short lists (main effect of other-list size), and that new lists were recalled worse when probed second rather than first (main effect of time of recall). Worse recall of lists probed second is evidence for output interference from the first-recalled list (e.g., Cowan, Saults, Elliott, & Moreno, 2002; Jones & Oberauer, 2013; Oberauer, 2003). The results were inconclusive on whether output interference is stronger (a) from a new list than from a chunk (Type Other x Time of Recall) and (b) from a longer list than from a shorter list (Size Other x Time of Recall) as the BFs in the two parallel analyses differed slightly (see Table 1).

We next ask whether the size of a chunk affects how well a new list is recalled. This analysis tests the hypothesis that a chunk imposes the same load on WM regardless of its size. If so, chunk size should have no effect on how well a new list can be maintained concurrently. However, chunk size could matter for output interference because larger chunks require output of more words. Our previous analysis yielded inconclusive evidence for output interference. Nevertheless, to exclude any potential effects of output interference, for this analysis, we zoomed in on trials in which the new list under

consideration was recalled first, thus removing any potential contribution of output interference. We compared recall accuracy of new lists in the conditions in which the other list was a chunk, and chunk size was either 2 or 4. If chunks reduce the load on WM independently of their size, recall accuracy of new lists should not differ as a function of chunk size. The BFs supported this claim (41.7 and 16.4 in favor of the null for new lists of size 2 and 4, respectively), which is strong evidence for the assumption that chunk size has no influence on how much capacity a chunk requires in WM.

Serial Recall of Chunked Lists. Chunked lists were recalled with a very high level of accuracy (*M* = .974). Recall of chunked lists varied as a function of other-list type (BF = 535), and other-list size (BF = 612). Chunks were remembered better together with another chunk (*M* = .985) than with a new list (*M* = .962), and better together with a list of size 2 (*M* = .985) compared to a list of size 4 (*M* = .962). There was no evidence for a main effect of size of the current chunk to be remembered (BF = 145 for the Null), and there was no substantial evidence for any higher-order interaction between these factors (all BFs < 3.2 for the Alternative).

Discussion

Experiment 1 used a design modeled after (Chen & Cowan, 2005) using a training phase to establish chunks of different sizes (2 or 4 words). It showed that storing a chunk reduced the load on WM: Remembering a random list of words was easier when another list to be remembered concurrently was a chunk than when it was another random list. Moreover, our data supported the claim that WM capacity is constrained by the number of chunks, and not by chunk size: Increasing the size of a chunk had no influence on remembering a random list of words when the random list was recalled first, whereas increasing the size of another random list did impair recall. The size of a chunk had no

influence on how much WM capacity it requires, in line with the prediction of the embedded-process theory of Cowan (2001).

(Chen & Cowan, 2005) suggested that people may only hold the first item of a chunk in WM, which allows retrieval of the subsequent items from LTM. This possibility exists for their earlier chunking experiments as well as for Experiment 1, because in these experiments all chunks consisted of unique elements, which could not be an element of any other chunk or new list. This peculiarity of the design has two potentially important implications. First, participants can use the first word of a chunk as retrieval cue to recall the entire chunk. Second, the chunk can already be detected upon presentation of the first element, so that any further elements do not even have to be encoded into WM, thereby avoiding the risk that they interfere with other information held in WM. Under these circumstances, any theory predicts that chunk size does not impact capacity, because what participants do is just remembering one word for chunks of any size.

We argue that chunks consisting of unique elements are not representative for naturally occurring chunks. In many situations, chunks do not start with unique elements. In these situations, it is not possible to stop encoding after the first item of the chunk has been presented. Further, remembering only the first item is not an advisable strategy either because other lists also start with the same item. Hence, application of the strategy to only remember the first item of a chunk is often not possible. To investigate the hypothesis that chunks reduce the load on WM irrespectively of chunk size, one has to use chunks with not-unique elements. This is what we did in the following experiments.

Experiment 2

In Experiment 2, we again tested whether there is a chunking benefit and whether it is independent of chunk size. However, we made sure that people cannot apply the simple strategy to only encode and remember the first item of a chunk. To that end we used chunks consisting of not-unique elements in Experiment 2.

We suspect that the chunking benefit may decrease when using not-unique elements that can also appear in other lists and other chunks. A chunk composed of not-unique elements cannot be detected before all its elements are encoded individually into WM. We explain the hypothesis by means of an example. Let us assume that a participant encodes a list in which the first two letters are F and B. The third letter can be either an I or a Q. The participant can only replace the individual representations of the letters with a chunk representation in the first case, in which the sequence is the well-known chunk FBI, but not in the second case, in which the sequence is FBQ. Replacement of the individual letters by a chunk is not possible until presentation of the third letter. Before that, the encoding processes in both cases are the same.

Let us first focus on the case that the three-letter sequence is presented before some other items are presented. After presentation of all items of the chunk no representation of other WM contents will be degraded. If the chunk representation is independent of chunk size, it follows that successful replacement of the individual items with the chunk representation reduces the load on WM independent of chunk size. Now let us consider the case where the three-letter sequence is presented after some other information has already been stored in WM. The memory representations of the earlier presented items will be degraded and remembered less well after presentation of the chunk when all items of the

chunk have to be encoded individually. This follows from any theory of WM, whether it ascribes the reason of forgetting in WM to decay with rehearsal counteracting decay, to a limited resource (as the theory of Cowan, in which the resource is a fixed number of discrete slots), or to interference: The representations of the earlier items have suffered more from decay, which could not be counteracted by rehearsal during encoding of the chunk, or they have received less of a share from the limited resource because it was redistributed to the individual chunk items, or they have suffered interference from encoding the individual items of the chunk. For these reasons, a chunk presented at the end of the entire memory set should be much less beneficial than a chunk presented at the beginning.

With regard to the above prediction about the serial position of a chunk, decay theories of WM differ subtly from resource and interference theories. A decay theory predicts a chunk benefit with a chunk in any position. This is because longer lists do not only take longer to be encoded, but also to be reactivated through rehearsal. A chunk at the end of the list does not reduce the amount of forgetting of other items during encoding, but it does reduce the amount of rehearsal needed. After detection of the chunk only the chunk representation has to be rehearsed, which takes less time than rehearsing all individual representations. Hence, even though a decay theory predicts that chunks help more when presented earlier in the list, it additionally predicts that chunks in any position still help memory for not-chunked items because a chunk takes less time to be rehearsed. In contrast, resource or interference theories do not predict any benefit from a chunk presented at the end of the entire memory set.

To summarize, all theories predict that the reduction of WM load by replacing multiple items by a single chunk is beneficial for subsequently encoded material, because capacity is freed up only for material encoded after that replacement. A decay-and-rehearsal

theory predicts additionally that chunks in any input position help memory for not-chunked items.

In Experiment 2, we assessed the effect of chunk size by comparing chunked lists of size 3 vs. singleton lists composed of a single letter. New lists in the present experiment consisted of 3 letters. As a consequence of this reduction of list size in comparison to Experiment 1, we had to increase the number of lists presented in every trial from two to three to circumvent potential ceiling effects. The training phase in the beginning of the experiment was omitted because we used well-known acronyms consisting of three consonants as chunks.

A prerequisite of Experiment 2 was that participants detected an acronym when presented in the very beginning of the list, but also when presented in the end of the list. (Portrat et al., 2016) showed that the latter may not be guaranteed. They observed that acronyms were recalled worse when presented in the middle or in the end of a list compared to the beginning of the list in a complex-span paradigm. It could be that the chunks were not detected in these conditions because it was difficult to anticipate when a chunk began and when it ended. It is impossible to investigate the beneficial effect of chunks on not-chunked information when the chunks are not detected. To maximize the chance that participants detected the chunks in any serial position of the memory set in the following two experiments, we broke the memory set into three clearly demarcated lists, each of which could be a chunk.

Method

Participants

Twenty university students (17 women; M ≈ 25 years old) participated in Experiment 2 for one session lasting approximately one hour. Participation was compensated with 15 Swiss Francs or partial course credit.

Materials

We constructed a pool of 30 known 3-consonant acronyms to serve as the chunked lists in Experiment 2. To create the 30 singleton lists, we took the first consonants of the chunked lists. To create the new lists of size 3, we shuffled the consonants of the chunked lists six times to construct 180 new lists. Shuffling sometimes re-created known acronyms. Therefore, individual consonants were exchanged with consonants from other new lists by hand to have all new lists differing from chunks. We deleted 15 new lists to obtain a total of 165 new lists because we needed 5.5 times as many new lists as chunks. The new lists were allowed to overlap with chunks in individual consonants at certain positions (i.e., position 1, 2, or 3) or in the beginning or in the ending pair (i.e., items 1 and 2 or items 2 and 3). To assess whether we were successful in creating chunked versus new lists, we compared those lists on two measures to assure that they only differed in overall familiarity but not in transition probabilities between consonants, which also influence short-term memory retention (Mayzner & Schoenberg, 1964). First, we compared the number of Google hits (restricted to Switzerland), which served as a measure of overall familiarity of the strings for the participants in the present experiments. If the chunks are more familiar, they should generate more Google hits than new lists. We computed $\log_{10}$ Google hits because of skew in the data and outliers in the upper range of the scale, and submitted them to a Bayesian t-test using the BayesFactor package (Morey & Rouder, 2014). The alternative hypothesis that

more hits were generated for chunks (*M* = 5.60) than for new lists (*M* = 4.19) was supported by a BF of 3.76×10^{41}. Second, we compared the case insensitive corpus frequencies (Heister et al., 2011, also $\log_{10}$ transformed) of the bigrams in the two types of lists with another t-test. A difference in this measure would suggest that some of the transitions between consonants are more frequent in one type of list. However, the BF was 2.1 for the null hypothesis, suggesting that chunks (*M* = 4.67) are not likely to differ from new lists (*M* = 4.47). Hence, our chunks as a whole were more familiar than the new lists but they were comparable in familiarity at the level of bigrams. All stimuli were presented twice across the experiment to control for familiarity within the experiment. The stimuli used in all experiments are available on the OSF webpage.

Procedure

In every trial, three rows, each consisting of three black box frames, were displayed on the screen. The rows were shown at the top, middle, and bottom of the screen. Each row served to present a memory list. Lists were presented in order from top to bottom. The letters composing each list were presented one-by-one (for 1 s each) from left to right. We were interested in the serial recall of new lists depending on the context they were presented in (together with a chunk or another new list), depending on chunk size (singleton lists vs. chunked list), and depending on the chunk position within a trial (beginning or the end of the trial). With this aim, we created five experimental conditions (see Figure 3). In all five conditions, two new lists of size three were presented. Only the remaining list varied between conditions. In the Singleton First and Singleton Last conditions, this critical list was a chunk of size one (singleton) presented in the upper row or in the bottom row, respectively. In both conditions, the singleton was presented in the third serial position in a row, and it was preceded by two leading blanks. In the Chunk First and Chunk Last

conditions, the critical list was a chunked list presented in the top row or in the bottom row, respectively. In the Baseline condition, all three lists were new lists of size three.

After presentation of all lists, the recall test started: Each row was cued to be recalled in left-to-right serial order. An empty black frame prompted participants to recall the list at the cued location (upper, middle, or lower row). For three-item lists or chunks, every consonant was sequentially cued with an individual empty frame. For singletons, only one empty frame at the respective third list position was presented as a cue. Participants confirmed responses with the enter key. In all five conditions, we controlled that each of the three rows was cued ten times as the first, ten times as the second, and ten times as the third list to be recalled. This resulted in 150 trials that were presented in a randomized order over the course of a single session.

Results

Serial Recall of Chunked Lists. As a manipulation check, we analyzed whether singletons and chunks were recalled better than new lists that were presented in comparable rows. Serial recall accuracies for all items in the five conditions are plotted in Figure 4 against item input position, and in Figure 5 for the three list types against row of presentation. It is clearly visible in both figures that singletons and chunks were recalled better than new lists. Surprisingly, three-element chunks tended to be recalled better than singletons. Across all input and output positions, the average recall accuracies were .74, .80, and .67 for singletons, chunks, and new lists, respectively. We computed pairwise comparisons of the means of the three list types in a Bayesian linear regression on proportion correct. There was overwhelming evidence for better recall of singletons vs. new lists (BF = 990), for better recall of chunks than new lists (BF = 2.9×10^{10}), but slight evidence against the apparent better recall of chunks than singletons (BF = 0.54).

Serial Recall of New Lists. Next, we focused on the impact of having a singleton or chunk in a trial upon recall of the other new lists. We performed two analyses – one on the data of the Singleton First and Chunk First conditions against the Baseline, and another one on the data of the Singleton Last and Chunk Last conditions against the Baseline. The first analysis focused on recall of new lists in in the middle and lower row, depending on whether a singleton, a chunk, or a new list was presented in the upper row in a trial. The second analysis focused on recall of new lists in the upper and middle row, depending on whether a singleton, a chunk, or a new list was presented in the lower row in a trial.

Chunk First. Mean serial recall accuracy of new lists when preceded by a singleton, a new list, or a chunk is shown in Figure 6a. We performed a Bayesian linear regression on these data with the variables row of presentation (middle vs. lower, zero-centered) and condition. The analysis focused on the question whether a chunk that is presented in the beginning of a trial helps retention of new (not-chunked) lists presented afterwards. We coded the three levels of the condition variable in terms of two simple-code contrasts: baseline vs. singletons and baseline vs. chunks. Given that lists were probed in random order, there was a variable number of lists recalled prior to recall of a given list. To control for the output interference from these prior recalls, we added the number of previous list recall attempts (also centered on zero) as a control variable into the regression analysis. The variable was entered as a continuous covariate into the analysis because previous work showed that output position affects memory approximately linearly (Oberauer, 2003). In addition to that, we wanted to know whether output interference differed between list types (e.g., is output interference of a chunk the same as of a new list?). Therefore, we entered two further zero-centered predictors into the model. The first coded whether, in the chunk condition, the secondly recalled list was preceded by the recall of a new list or a

chunk. The second coded whether, in the singleton condition, the secondly recalled list was preceded by the recall of a new list or a singleton.

Because some of the chunks may not have been familiar to all participants, we performed the same analysis only for those trials in which singletons and chunks were recalled correctly (Figure 6b). When chunks were recalled correctly, we can be more confident that participants actually recognized and remembered them as chunks, and also that there were no trade-offs between maintenance of the chunk or singleton and new lists. Posterior means, 95% HDIs, and BFs of the effects are shown in Table 2.

The evidence for the effects of interest was qualitatively the same across the two sets of analyses shown in Table 2. The evidence for better recall of new lists in the presence of a singleton or chunk was decisive. However, contrasting the effects for singleton and chunk showed that memory for new lists was better in the context of a singleton than in the context of a chunk. There was no evidence for a main effect of row of presentation (middle vs. lower) on accuracy. The benefit of having a singleton in the upper row was larger in the middle row than in the lower row (Singleton x Row interaction), but the evidence was not strong. There was no evidence for the chunk benefit to differ between the middle and the lower row. As expected, the parameter for the number of previously recalled lists was negative. This shows that every additional list that has been recalled previously leads to worse memory for later recalled lists. Finally, there is moderate to strong evidence that output interference does not differ between different list types (Recalled Chunk Before and

Chunk Last. Next, we analyzed the data of the Singleton Last and Chunk Last conditions against the Baseline, which addresses the question whether a chunk helps retention of previously presented new lists. We ran the same analysis as for Chunk First (see Table 3). The data are shown in Figure 7. There was only evidence for a benefit for recall of

new lists in the singleton condition, but not in the chunk condition. There was no difference in accuracy across conditions between the upper and the middle row. There was no evidence that having a singleton or a chunk in the lower row differentially affected memory for new lists in the upper or in the middle row (Singleton x Row and Chunk x Row interactions).

Discussion

Chunks were recalled better than random lists, demonstrating that our manipulation worked. Chunks even tended to be recalled better than singletons. The main finding of Experiment 2 was that chunks improved recall of later presented information but not recall of earlier presented information compared to random lists. This finding is consistent with theories explaining WM capacity by the allocation of a limited resource or by interference between representations encoded into WM. The fact that chunks in the lower row did not help recall of the preceding new lists (in the upper and middle row) at all is partially at odds with decay-and-rehearsal theories, as will be explained in the following.

All three theoretical perspectives assume that the Baseline condition and the Chunk Last condition do not differ until all items within a trial have been presented. The chunk itself can only be recognized after all items of the chunk have been presented (Bower & Springston, 1970). For instance, consider a resource theory with discrete resources (a.k.a. slots) sufficient to maintain four chunks (Awh, Barton, & Vogel, 2007; Cowan, 2001; Zhang & Luck, 2008). Immediately after presentation of the last item in the lower row, four items are held in four slots, and any additional items could not be accommodated. Even though the last three items can now be replaced by a single chunk, earlier presented items cannot be recovered because they dropped out of WM earlier. Therefore, memory for the first and second list should not benefit from recognition of a chunk in the lower row. In contrast,

when the upper list is a chunk, a chunk representation can be retrieved from LTM, and the three individual item representations can be dropped from WM. As a consequence, participants have two more free slots in the Chunk First condition than in the Baseline condition that they can use to maintain subsequent items, resulting in better memory for the middle and lower lists.

An interference account (Oberauer & Lewandowsky, 2008) states that forgetting is due to items in WM interfering with each other. Right after seeing the last item, interference should be the same in the Chunk Last condition and in the Baseline condition because a chunk representation can only be retrieved from LTM after all individual chunk items have been encoded into WM. In contrast, when a chunk is presented first, the representation of the chunk is loaded into WM, while the individual representations of its elements can be removed in one sweep ("wiping" WM, Ecker et al., 2014). After that, the only representation in WM is the chunk representation. Moreover, the individual representations initially encoded did not damage any other information already stored in WM. In contrast, when the chunk is presented last, the chunk is only retrieved from LTM after all nine letters of the three lists have been encoded into WM. At this point in time, any damage done to previously encoded items is difficult to repair because the items of the chunk have to be removed individually. Selective removal of individual items is slower than removal of all contents of WM (Ecker et al., 2014) and arguably more error-prone.

Although the benefit of chunks is also assumed to be larger when presented first, a decay-and-rehearsal theory nevertheless predicts that chunks presented last should help memory for earlier presented lists. This is because a single chunk representation requires less time to be rehearsed than the representations of three individual consonants. Hence, even though a chunk should help more for not-chunked information when presented first, it

is also assumed to help when presented last. The results of Experiment 2 do not support this claim.

A singleton in the upper row improved memory for subsequent lists more than a chunk in the upper row. This finding is challenging for the assumption in the embedded-process theory that the amount of WM capacity a chunk requires in WM is independent of chunk size (Cowan, 2001).

A potential critique regarding the smaller benefit for chunks than singletons could be that we used known acronyms as chunks. It may be that not all participants were familiar with all acronyms. Therefore, the dependency of the size of the chunk benefit on chunk size may be because participants did not perfectly know the acronyms. However, the fact that the acronyms tended to be recalled even better than singletons suggests that this was likely not the case.

A further critique is that the leading blanks in the Singleton Last condition before presentation of the singleton allowed participants to strengthen memory for the previously presented lists. The same is not possible for chunks because there was no free time in-between presentation of the middle and lower list. In addition, the blanks in between the middle list and the singleton may have made the lists temporally more distinct (Brown, Neath, & Chater, 2007). To control for these possibilities, we replicated Experiment 2 (n = 20) but changed the leading blanks to ending blanks following presentation of the singleton in the Singleton Last condition. This removes any break in-between presentation of the middle and the lower row, thereby removing any effect of temporal distinctiveness and prolonged encoding of the middle-row list before encoding of the lower-row singleton. This modification did not change any of the main findings reported in Experiment 2, namely that

singletons but not chunks improved memory for new lists encoded before. The results of this experiment are available in the online supplementary materials.

In the control experiment we observed the same tendency as in Experiment 2 for chunks to be remembered better than singletons. When analyzing the data of both experiments together, there was strong evidence (BF = 15) that chunks were remembered better than singletons. This is a further challenging finding for the embedded-process theory because memory for chunks should be independent of chunk size. At least two interpretations of this finding seem viable. First, chunks are semantically more distinct in a trial than singletons. Representations of the former may have richer associations in LTM because chunks were known acronyms. Hence, visual or semantic representations in LTM may assist recall in addition to the chunk representation in WM. Singletons are certainly less distinct in that sense because they were also the individual elements of the other lists. Second, representations of the individual items forming the chunk may linger in WM due to incomplete removal. In combination with the chunk representation, they may aid recall of the chunk, while at the same time diminishing the chunking benefit for recall of new lists.

A final critique pertains to the type of representations used in Experiment 2 and the control experiment. (Chen & Cowan, 2009) distinguish between *central* and *phonological* storage, and only central storage is assumed to be limited by a fixed number of chunks. AS is meant to prevent maintenance of phonological representations. Chen and Cowan observed recall of a constant number of chunks only with AS but not without AS. These authors argued that any influence from phonological storage obscures a capacity limit in terms of chunks. It could be argued that participants in our first two experiments remembered a chunk not only by the chunk representation but also by the phonological representations of the individual items. If that was the case, remembering the new lists may have been more difficult in the

chunk condition than in the singleton condition, for example, due to competition for rehearsal, or due to domain-specific interference between phonological representations (see (Thalmann & Oberauer, 2017)). This difference should disappear if participants only used central storage to remember the chunks and singletons.

Experiment 3

In Experiment 3, we added AS to restrict the use of phonological representations. We tested whether the benefit of singletons was still larger than the benefit of chunks when phonological storage is prevented. The logic was the following: It is possible that in Experiment 2 singletons helped more than chunks because of the partial reliance on phonological representations, which are more complex, and take longer to rehearse, for three-letter acronyms than for single letters. If that is the case, the differential benefit of singletons vs. chunks should disappear under AS in Experiment 3. However, if we still observe the singleton benefit to be larger than the chunk benefit, we can be confident that the fact that chunks size matters is independent of phonological length and complexity.

Methods

Participants, Materials, and Procedure

Twenty university students (16 women; M ≈ 24 years old) participated in Experiment 3 for one session lasting approximately one hour. Participation was compensated with 15 Swiss Francs or partial course credit. The materials and the procedure were exactly the same as in Experiment 2, except that participants engaged in AS. Participants started to articulate continuously “ba bi bu” at a self-chosen rate before the stimuli were presented and stopped to do so when the first recall cue appeared on the screen.

Results

Serial Recall of Chunked Lists.

Memory performance in the five experimental conditions is plotted against item input position in Figure 8 and memory performance for the three list types is plotted against row of presentation in Figure 9. We tested as a manipulation check whether chunks and singletons were recalled more accurately than new lists. On average, recall accuracy was .71, .64, and .47 for singletons, chunks, and for new lists, respectively. Again, the evidence was compelling that singletons were remembered better than new lists (BF = 4.1e+11), and that chunks were remembered better than new lists (BF = 320’185). In contrast to Experiment 2 and the control experiment, chunks were not remembered better than singletons (BF = 0.16). Comparing the average recall accuracies for the three list types with the two previous experiments shows that adding AS decreased memory especially for chunks and new lists, but hardly for singletons.

Serial Recall of New Lists: Chunk First.

We tested with the same Bayesian regression model as in Experiment 2 whether new lists were recalled more accurately than in the control condition when a chunk or a singleton was presented in the upper row. In Figure 10 it appears that both chunks and singletons improved memory for later presented new lists. The analyses (see Table 4) showed, however, that only singletons but not chunks improved memory for later presented new lists credibly. Most importantly for the current purpose, the benefit on later presented new lists was credibly larger for singletons than for chunks. There was no evidence that the effect of having a singleton or chunk in the upper row differed between the middle and lower row (Singleton x Row, Chunk x Row). Finally, there was decisive evidence for output interference: The number of previously recalled lists had a deteriorating effect on new list recall. There

was again no compelling evidence that output interference from recalling a singleton or a chunk damaged memory for subsequently tested lists less than output interference from recalling a new list. This suggests that output interference happens at the level of lists and not at the level of individual singletons or chunks.

Serial Recall of New Lists: Chunk Last.

Next, we tested whether having a singleton or a chunk in the lower row benefitted recall accuracy of earlier presented new lists. Figure 11 shows better recall on average for new lists compared to the control condition when a singleton followed, but not when a chunk followed. The statistical analysis (see Table 5) confirmed that impression by showing decisive evidence for the comparison Singleton vs. Baseline, but strong evidence for the Null for the comparison Chunk vs. Baseline. The effect of having a singleton or chunk presented in the lower row did not differ between the upper and the middle row (Singleton x Row, Chunk x Row) and there was compelling evidence that having a singleton in the lower row was more beneficial than having a chunk in the lower row. The results regarding output interference corroborate the previous results: The number of previously recalled lists decreased memory, but type of list did not matter (Recalled Chunk Before and Recalled Singleton Before).

Discussion

The main question of Experiment 3 was whether having a singleton in the beginning of a trial still benefitted memory for new lists more than a chunk when participants were required to perform AS. According to (Chen & Cowan, 2009) only central storage is limited by a fixed number of chunks, and AS forces participants to rely predominantly on central storage. However, even with AS we still observed that the singleton benefit was larger than the chunk benefit. This result further supports our conclusion that the size of a chunk is

important in determining its beneficial effect when chunks cannot be maintained by remembering their first element. Likely, the chunk-size effect cannot be attributed to output interference because first recalling a chunk or a singleton decreased memory for about the same amount as recalling a new list. This finding confirms the prediction by (Farrell, 2012) according to which output interference happens at the level of lists (i.e., clusters). The main difference to Experiment 2 and the control experiment was that adding AS reduced memory for chunks and new lists, but hardly for singletons. At the moment, we can only speculate why this happened. For example, it could be that participants prioritized retention of singletons over retention of chunks and new lists.

Experiment 4

Experiments 2 and 3 showed that having a chunk in a trial helped retention of new lists only when the chunk was presented in the upper row, but not in the lower row. We assume that a chunk reduces the load in a WM task only after it has been recognized as a chunk. Only then can participants replace the representations of the individual items with a chunk representation. Clearly, recognizing a chunk in the lower row is not more difficult than in any other row, which is shown by the superior recall of chunks presented in this row compared to new lists in Experiments 2 and 3 (see Figures 4 and 5 and Figures 8 and 9). Apparently the reduction of load afforded by the chunk cannot repair the damage that has been added to representations of previously encoded lists. However, given that the previous experiments only assessed the effects of chunks presented at the beginning or at the end of the memory set (i.e., upper and lower row), we still do not know whether a chunking benefit is observed if a chunk appears mid-way through the trial.

Therefore, the main question addressed in Experiment 4 was how the beneficial effect of chunks depends on their serial position. To attain this goal, we focused exclusively on the chunk conditions (dropping the singleton conditions), and allowed the chunk to appear in all three rows. The previous experiments render two hypotheses plausible. The first possibility is that chunks only yield a benefit for lists encoded after the chunk has been presented, but not for previously encoded lists. This hypothesis rests on the assumption that chunking can free capacity for encoding subsequent items, but cannot undo the damage to already encoded items. The second possibility is that chunking also helps preceding lists as long as the damage done to them is only mild (i.e., only moderate interference, or reduction of resources, or decay), so that these lists can still be repaired after the load on WM has been reduced. The novel condition, in which a chunk was presented in the middle row, allowed us to test these two hypotheses.

Methods

Participants

Thirty-two university students (22 women; $M \approx 23$ years old) participated in one 1-hour session in exchange for 15 Swiss Francs or partial course credit.

Materials

We used the same set of 30 chunks as in Experiments 2 and 3. Because we dropped the two singleton conditions we only required three times as many new lists as chunks even though chunks could appear now in all three rows. We created a pool of 90 new lists by shuffling the letters of the chunks three times. The algorithm checked that no consonant was used twice within the same list. We changed one or the other letter between new lists manually because chunks were sometimes re-created via shuffling. We again compared chunks and new lists on overall familiarity measured as Google hits (restricted to

Switzerland) and on bigram frequency. The Bayesian t-test on $\log_{10}$ Google hits showed decisive evidence (BF = 1.34×10^{31}) that chunks (*M* = 5.62) were more familiar than new lists (*M* = 4.02). Another Bayesian t-test on $\log_{10}$ bigram frequencies showed substantial evidence for the null hypothesis (BF = 6.16) that chunks (*M* = 4.21) did not differ from new lists (*M* = 4.20). All lists were used three times in Experiment 3 and they are available on the OSF webpage.

Procedure

In total, there were four conditions. In all four conditions, three 3-consonant lists were presented sequentially on the screen from top to bottom as in Experiment 2. In the Baseline condition, three new lists were presented for encoding. In the remaining conditions, a chunk was presented either in the upper, middle, or lower row. Every list item was presented for 1 s. There was no time between presentations of two consonants within or between lists. Participants were required to recall the three lists immediately after presentation of the ninth consonant. In all four conditions, the lists in all rows were probed ten times to be recalled first, second, and third. Recall and scoring were the same as previously.

Results

Serial Recall of Chunked Lists. First, we compared serial recall accuracy for chunks and new lists (shown in Figures 12 and 13) with a Bayesian linear regression as a manipulation check. This time, we used the data from all three rows because chunks and new lists could be presented in any row. The analysis indicated that chunks (*M* = .81) were remembered better than new lists (*M* = .62), which was supported with a BF = 1.5×10^{24}.

Serial Recall of New Lists: Chunk Before. Next, we analyzed recall of new lists as a function of whether or not a chunk appeared in a preceding row. Figure 14a shows the data

of all trials, and Figure 14b shows data conditioned on the correct recall of the chunk. Serial recall accuracy was analyzed in a Bayesian linear regression using as independent variables row of presentation (middle vs. lower, zero-centered) and condition (Baseline, chunk in the upper row, and chunk in the middle row). The latter variable was entered into the regression as two simple-coded variables using the Baseline condition as the reference category. The first contrast, Chunk in Upper Row, compared the Baseline condition to lists that were preceded by a chunk in the upper row; the second contrast, Chunk in Middle row, compared the Baseline condition to lists that were preceded by a chunk in the middle row (which could only happen for lists in the lower row). In Figure 14 it is visible that a new list was recalled more accurately when it was preceded by a chunk in the upper or middle row. There was substantial to strong evidence (see Table 6) that a chunk in the upper row increased memory more for lists in the middle row than the lower row.

Serial Recall of New Lists: Chunk After. We analyzed recall accuracy of new lists that were followed by a chunk and compared it with trials in which only new lists were presented (see data on Figures 15a and 15b) using a Bayesian linear regression. Row of presentation (again zero-centered) and condition were used as independent variables. The latter variable was entered as two simple-coded variables. The first variable, Chunk in Middle Row, compared the Baseline condition to the condition with a chunk in the middle row. The second variable, Chunk in Lower Row, compared the Baseline condition to the condition with a chunk in the lower row. The only constellation in which a chunk helped memory for a previously presented list was when the chunk appeared in the middle row: in this case, memory for the list in the upper row improved compared to the Baseline condition. The BFs and the HDIs, which are shown in Table 7, indicate that the beneficial effect of a chunk in the middle row was credible. The evidence was strongly against a beneficial effect of a chunk in

the lower row, replicating the result from Experiment 2. There was also evidence against the two-way interaction. Together, the two analyses of memory for new lists depending on chunk position were in line with Experiments 2 and 3. They supported the claim that the detection of chunks helps remembering new lists as long as WM has not been heavily loaded.

Discussion

Experiment 4 focused on the question at what list positions chunks helped remembering new lists in a trial. By varying the serial input position of the chunk in the trial we evaluated whether a chunk only helps memory for lists that follow it (because it frees capacity to encode those lists) or whether a chunk can help lists that preceded it (because it frees capacity to re-establish degraded representations in WM).

As in the previous experiments, chunks improved memory for the following lists. This was also the case for chunks presented in an intermediate serial position in the memory set (i.e., in the middle row). Presenting a chunk in the middle row also improved memory for the preceding list. After presentation of the middle–row list, WM is already loaded with six consonants. At that point, the representations of the consonants in the first list (upper row) have been degraded from encoding the consonants in the second list (middle row). Nevertheless, recognizing the chunk in the second list helped memory for the first list. If we assume that representations of different items interfere with each other, the representations of the first list will be distorted by the subsequent encoding of the items of the second list. These distorted representation have to be disambiguated to be recalled, a process which is called redintegration (Lewandowsky, 1999). If the second list items can be replaced by a chunk, the first list items can still be retrieved and redintegrated before they are further distorted by the third list. In addition, if the individual items of the chunked list

can be replaced by a chunk representation, the total amount of interference in WM is reduced. Hence, better memory for the first list could be explained if we think of participants redintegrating successfully the first list after recognizing the chunk, thereby reconstructing the representations of the first list's items. Redintegration will be less successful in the Chunk Last condition because more interference between representations has happened after encoding nine items.

General Discussion

The goal of the present series of experiments was to examine the relation between WM and LTM in serial recall tasks using sequential presentation of stimuli. We evaluated how WM can make use of information stored in LTM by recoding several stimuli into a familiar unit, a process known as chunking.

Evidence that Chunking frees Capacity

Previous work showed that the amount of information recalled in WM tasks could be well described by a fixed number of chunks (Chen & Cowan, 2005, 2009; Cowan et al., 2004). However, these studies left open whether chunks help because they free capacity in WM or because LTM assist reconstruction of chunks at recall (Hulme et al., 1991; Hulme & Roodenrys, 1995). To test whether chunking frees capacity for other, not-chunked information, we tested the chunking benefit on not-chunked information. We consistently observed such a benefit across all experiments, confirming that chunking frees WM capacity.

Chunk Size Matters

If WM capacity is a limit on the number of chunks, chunk size should have no effect on how much capacity a chunk consumes (Chen & Cowan, 2005, 2009). Therefore, our next aim was to test whether the chunking benefit depended on chunk size. It did not in

Experiment 1, but it did in Experiments 2, 3, and the control experiment. In the latter experiments, we constructed the chunks so that participants cannot restrict encoding and maintenance to the first element of a chunk. In contrast, the chunks used in Experiment 1 allowed one to use this strategy, because the first element was unique to every chunk. With the assumption of that strategy, any theory of WM predicts that chunk size has no influence on the chunking benefit. This follows because participants are not required to encode and maintain the whole chunk, but only a single item. When participants were required to encode all elements of a chunk, as in Experiment 2, there was a smaller chunking benefit for three-element chunks compared to singletons.

The constant capacity in terms of a fixed number of chunks reported in earlier studies (Chen & Cowan, 2005, 2009; Cowan et al., 2004) may to some degree be due to the material that allowed the application of the above mentioned memory strategy. We cannot make this attribution with confidence, however, because the comparison between our Experiment 1 – enabling this strategy – to our Experiments 2, 3, and 4 – not enabling the strategy – is confounded with other differences, in particular pertaining to the materials and the size of the memory sets. Moreover, there are arguably other factors, apart from strategy differences, that contribute to the different results when using chunks with unique or not unique elements. For example, when using chunks with not-unique elements, similarity between two chunks sharing some elements, and between chunks and singletons, is larger than when chunks are composed of unique elements. Similarity between list items increases the chance of confusing them, thereby decreasing serial recall performance (Conrad, 1964; Saito, Logie, Morita, & Law, 2008).

Could similarity-based confusion explain why singletons led to better memory for other items than three-letter chunks did? For such an explanation to work, we would have to

assume that not-chunked items are confused more often with three-letter chunks than with singletons. If that were the case, memory for chunks should be worse than memory for singletons. The opposite was the case in Experiment 2 (and the accompanying control experiment), not supporting a similarity-based confusion account. Such an account could, however, be tested experimentally by varying whether chunks and singletons share elements with new lists. We leave this possibility open to future research.

Could other forms of interference – apart from confusion – explain the differential benefits of singletons and three-letter chunks? If we assume that the representations of three-letter chunks are more complex than those of singletons (i.e., containing more features or components, such as more phonemes or more letters), then they could interfere more strongly with other items by distorting their representations – a mechanism known as interference by superposition (Oberauer, Lewandowsky, Farrell, Jarrold, & Greaves, 2012).

Any explanation in terms of factors leading to different amounts of interference between representations takes a step away from the notion that WM capacity is limited in terms of a fixed number of chunks. According to such a model – sometimes referred to as "slot model" – performance should be limited only by the number of chunk representations. Variables such as their similarity or complexity should not matter. It is still possible that there is a core capacity limit in terms of chunks, which is obscured by additional mechanisms affecting memory performance. We ruled out one of them, phonological maintenance and rehearsal, in Experiment 3, but there is an infinite number of other auxiliary mechanisms – including interference – that could be added to a discrete-capacity model (for an example of such an augmented model, see (Cowan, Rouder, Blume, & Saults, 2012)). The more such additional mechanisms are invoked, however, the more we need to ask whether the

additional mechanisms are not sufficient to explain all extant data on their own, without the assumption of a core capacity limit (Navon, 1984).

Chunking is more beneficial for subsequent than for earlier contents of WM

The chunking benefit interacted with serial position of the chunk. Whereas a chunk benefitted other not-chunked information when presented as the first or second list, there was no such benefit when presented as the third list. This is in agreement with the results by (Portrat et al., 2016)Figure 8, p. 431). Although Portrat and colleagues did not test the benefit of chunks on not-chunked information formally, their figure suggests that memory for subsequent letters improved when the preceding letters formed a chunk rather than a random set. We assume that chunks only help retention of other items after a compact chunk representation has been retrieved from LTM, and the representations of the individual items have been removed from WM. These processes can only take place after all elements of a chunk have been presented. Up to that point, the representations already in WM will be damaged by the temporary maintenance of the individual elements of the chunk. When the damage to representations of earlier encoded items is only mild, removal of the representations of the individual chunk elements allows reparation of the damage. However, when the damage is severe, reparation is not possible anymore. This is why chunks presented in the last list position did not lead to a chunking benefit. As we used serial presentation of all stimuli, it is possible that the effect of list position is diminished (a) when all items of a list are presented simultaneously onscreen (allowing detection and encoding of the full chunk at once) or (b) when a single chunk representation for the whole list can be retrieved after presentation of each individual item of the list (e.g., ice – cream – man[3]). A decay-and-reactivation theory of WM additionally predicted that chunks help in any list position, because they require less time to be rehearsed in the retention interval than new lists. However, we did not find evidence for this prediction, consistent with evidence that verbal representations do not decay in WM (Oberauer & Lewandowsky, 2013, 2014).

[3] We would like to thank Nelson Cowan for this suggestion.

Conclusion

Chunks reduce the load on WM, thereby improving memory for other information maintained concurrently. The load by a three-letter chunk, however, still exceeds that from a single letter. Hence, a fixed capacity in terms of number of chunks in WM cannot alone explain the chunking benefit. We propose that the chunking benefit results from the following steps: After the individual items have been encoded into WM, potentially interfering with already encoded representations, a matching chunk representation can be detected in LTM. This representation is retrieved from LTM and encoded into WM, which allows removal of the representations of the individual elements of the chunk. The ensuing reduction of load on WM facilitates subsequent encoding of further information into WM. In contrast, information encoded before the chunk suffers from the initially high load, and the subsequent reduction of load enables only limited repair of that damage.

References

Awh, E., Barton, B., & Vogel, E. K. (2007). Visual Working Memory Represents a Fixed Number of Items Regardless of Complexity. *Psychological Science*, *18*(7), 622–628. https://doi.org/10.1111/j.1467-9280.2007.01949.x

Bower, G. H., & Springston, F. (1970). Pauses as recoding points in letter series. *Journal of Experimental Psychology*, *83*(3, Pt.1), 421–430. https://doi.org/10.1037/h0028863

Brainard, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision*, *10*(4), 433–436. https://doi.org/10.1163/156856897X00357

Brener, R. (1940). An experimental investigation of memory span. *Journal of Experimental Psychology*, *26*(5), 467. https://doi.org/10.1037/h0061096

Brown, G. D. A., Neath, I., & Chater, N. (2007). A temporal ratio model of memory. *Psychological Review*, *114*(3), 539–576. https://doi.org/10.1037/0033-295X.114.3.539

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, *4*(1), 55–81. https://doi.org/10.1016/0010-0285(73)90004-2

Chen, Z., & Cowan, N. (2005). Chunk Limits and Length Limits in Immediate Recall: A Reconciliation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*(6), 1235–1249. https://doi.org/10.1037/0278-7393.31.6.1235

Chen, Z., & Cowan, N. (2009). Core verbal working-memory capacity: The limit in words retained without covert articulation. *The Quarterly Journal of Experimental Psychology*, *62*(7), 1420–1429. https://doi.org/10.1080/17470210802453977

Conrad, R. (1964). Acoustic Confusions in Immediate Memory. *British Journal of Psychology*, *55*(1), 75–84. https://doi.org/10.1111/j.2044-8295.1964.tb00899.x

Cowan, N. (2001). The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *The Behavioral and Brain Sciences*, *24*(1), 87–114; discussion 114-185.

Cowan, N., Rouder, J. N., Blume, C. L., & Saults, J. S. (2012). Models of verbal working memory capacity: What does it take to make them work? *Psychological Review*, *119*(3), 480–499. https://doi.org/10.1037/a0027791

Cowan, N., Saults, J. S., Elliott, E. M., & Moreno, M. V. (2002). Deconfounding Serial Recall. *Journal of Memory and Language*, *46*(1), 153–177. https://doi.org/10.1006/jmla.2001.2805

Cowan, N., Zhijian Chen, & Rouder, J. N. (2004). Constant Capacity in an Immediate Serial-Recall Task. *Psychological Science (Wiley-Blackwell)*, *15*(9), 634–640. https://doi.org/10.1111/j.0956-7976.2004.00732.x

Ecker, U. K. H., Oberauer, K., & Lewandowsky, S. (2014). Working memory updating involves item-specific removal. *Journal of Memory and Language*, *74*, 1–15. https://doi.org/10.1016/j.jml.2014.03.006

Farrell, S. (2012). Temporal clustering and sequencing in short-term memory and episodic memory. *Psychological Review*, *119*(2), 223–271. https://doi.org/10.1037/a0027371

Gobet, F., & Clarkson, G. (2004). Chunks in expert memory: Evidence for the magical number four … or is it two? *Memory*, *12*(6), 732–747. https://doi.org/10.1080/09658210344000530

Hulme, C., Maughan, S., & Brown, G. D. A. (1991). Memory for familiar and unfamiliar words: Evidence for a long-term memory contribution to short-term memory span. *Journal of Memory and Language*, *30*(6), 685–701. https://doi.org/10.1016/0749-596X(91)90032-F

Hulme, C., & Roodenrys, S. (1995). The role of long-term memory mechanisms in memory span. *British Journal of Psychology*, *86*(4), 527.

Jones, T., & Oberauer, K. (2013). Serial-position effects for items and relations in short-term memory. *Memory*, *21*(3), 347–365. https://doi.org/10.1080/09658211.2012.726629

Kass, R. E., & Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association*, *90*(430), 773–795. https://doi.org/10.2307/2291091

Kruschke, J. K. (2011). *Doing Bayesian Data Analysis: A Tutorial with R and BUGS*.

Lee, M. D., & Wagenmakers, E.-J. (2014). *Bayesian Cognitive Modeling: A Practical Course*. Cambridge ; New York: Cambridge University Press.

Lewandowsky, S. (1999). Redintegration and Response Suppression in Serial Recall: A Dynamic Network Model. *International Journal of Psychology*, *34*(5/6), 434–446. https://doi.org/10.1080/002075999399792

Mayzner, M. S., & Schoenberg, K. M. (1964). Single-letter and digram frequency effects in immediate serial recall. *Journal of Verbal Learning and Verbal Behavior*, *3*(5), 397–400. https://doi.org/10.1016/S0022-5371(64)80008-6

Morey, R. D., Hoekstra, R., Rouder, J. N., Lee, M. D., & Wagenmakers, E.-J. (2016). The fallacy of placing confidence in confidence intervals. *Psychonomic Bulletin & Review*, *23*(1), 103–123. https://doi.org/10.3758/s13423-015-0947-8

Morey, R. D., & Rouder, J. N. (2014). BayesFactor: Computation of Bayes factors for common designs (Version 0.9.7). Retrieved from http://cran.at.r-project.org/web/packages/BayesFactor/index.html

Navon, D. (1984). Resources—a theoretical soup stone? *Psychological Review*, *91*(2), 216–234. https://doi.org/10.1037/0033-295X.91.2.216

Oberauer, K. (2003). Understanding serial position curves in short-term recognition and recall. *Journal of Memory and Language*, *49*(4), 469–483. https://doi.org/10.1016/S0749-596X(03)00080-9

Oberauer, K., & Lewandowsky, S. (2008). Forgetting in immediate serial recall: Decay, temporal distinctiveness, or interference? *Psychological Review*, *115*(3), 544–576. https://doi.org/10.1037/0033-295X.115.3.544

Oberauer, K., & Lewandowsky, S. (2013). Evidence against decay in verbal working memory. *Journal of Experimental Psychology: General*, *142*(2), 380–411. https://doi.org/10.1037/a0029588

Oberauer, K., & Lewandowsky, S. (2014). Further evidence against decay in working memory. *Journal of Memory and Language*, *73*, 15–30. https://doi.org/10.1016/j.jml.2014.02.003

Oberauer, K., Lewandowsky, S., Farrell, S., Jarrold, C., & Greaves, M. (2012). Modeling working memory: An interference model of complex span. *Psychonomic Bulletin & Review*, *19*(5), 779–819. https://doi.org/10.3758/s13423-012-0272-4

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: transforming numbers into movies. *Spatial Vision*, *10*(4), 437–442. https://doi.org/10.1163/156856897X00366

Plummer, M. (2003). *{JAGS}: A program for analysis of {Bayesian} graphical models using {Gibbs} sampling*. Presented at the Proceedings of the 3rd International Workshop on Distributed Statistical Computing.

Portrat, S., Guida, A., Phénix, T., & Lemaire, B. (2016). Promoting the experimental dialogue between working memory and chunking: Behavioral data and simulation. *Memory & Cognition*, *44*(3), 420–434. https://doi.org/10.3758/s13421-015-0572-9

Saito, S., Logie, R. H., Morita, A., & Law, A. (2008). Visual and phonological similarity effects in verbal immediate serial recall: A test with kanji materials. *Journal of Memory and Language*, *59*(1), 1–17. https://doi.org/10.1016/j.jml.2008.01.004

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Linde, A. van der. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *64*(4), 583.

Thalmann, M., & Oberauer, K. (2017). Domain-specific interference between storage and processing in complex span is driven by cognitive and motor operations. *The Quarterly Journal of Experimental Psychology*, *70*(1), 109–126. https://doi.org/10.1080/17470218.2015.1125935

Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems ofp values. *Psychonomic Bulletin & Review*, *14*(5), 779–804. https://doi.org/10.3758/BF03194105

Wetzels, R., Raaijmakers, J. G. W., Jakab, E., & Wagenmakers, E.-J. (2009). How to quantify support for and against the null hypothesis: A flexible WinBUGS implementation of a default Bayesian t test. *Psychonomic Bulletin & Review*, *16*(4), 752–760. https://doi.org/10.3758/PBR.16.4.752

Zhang, W., & Luck, S. J. (2008). Discrete fixed-resolution representations in visual working memory. *Nature*, *453*(7192), 233–235. https://doi.org/10.1038/nature06860

Tables

Table 1

Posterior Means, Lower and Upper Boundaries of the 95% HDIs, and the BFs for the Parameters of the Linear Model fitted to the Data of Experiment 1.

		Measure			
			95% HDI		
	Effect	Posterior Mean	Lower Bound	Upper Bound	Bayes Factor
Set Size = 2					
	Type Other	0.08	0.05	0.11	249222
	Size Other	-0.13	-0.16	-0.10	3.90E+16
	Time of Recall	-0.11	-0.13	-0.08	7.50E+10
	Type Other x Size Other	0.19	0.14	0.25	1.10E+09
	Type Other x Time of Recall	0.06	0.01	0.12	4.60E-01
	Size Other x Time of Recall	-0.16	-0.21	-0.10	5.81E+05
	Three Way	0.19	0.08	0.30	2.90E+01
Set Size = 4					
	Type Other	0.18	0.16	0.21	1.30E+36
	Size Other	-0.11	-0.14	-0.08	1.20E+11
	Time of Recall	-0.11	-0.13	-0.08	6.20E+10
	Type Other x Size Other	0.17	0.12	0.22	3.72E+06
	Type Other x Time of Recall	0.09	0.03	0.14	5.50E+00
	Size Other x Time of Recall	-0.07	-0.12	-0.01	7.30E-01
	Three Way	0.08	-0.03	0.19	0.24

Table 2

Posteriors means, 95% HDIs, and the BFs of the parameters of the linear regression model fitted to the data of Experiment 2 presented in panels a and b of Figure 6, respectively.

		Measure			
		Posterior Mean	95% HDI		Bayes Factor
Chunk First	**Effect**		Lower Bound	Upper Bound	
All Trials	Chunk (vs. Baseline)	0.09	0.06	0.12	187’566.00
	Singleton (vs. Baseline)	0.21	0.18	0.24	2.50E+36
	Row	0.02	0.00	0.05	0.10
	Chunk x Row	-0.06	-0.13	0.00	0.41
	Singleton x Row	-0.09	-0.15	-0.03	2.30
	Singleton vs. Chunk	0.12	0.09	0.15	2.70E+10
	Nr. Previous Recallled Lists	-0.12	-0.14	-0.11	2.00E+50
	Recalled Chunk Before	-0.03	-0.10	0.05	0.08
	Recalled Singleton Before	0.08	0.00	0.15	0.41
Chunks and Singletons Correct	Chunk (vs. Baseline)	0.10	0.06	0.13	92’066.00
	Singleton (vs. Baseline)	0.22	0.18	0.25	4.30E+30
	Row	0.02	0.00	0.05	0.09
	Chunk x Row	-0.06	-0.13	0.00	0.40
	Singleton x Row	-0.09	-0.15	-0.03	2.20
	Singleton vs Chunk	0.12	0.08	0.16	3.19E+05
	Nr. Previous Recallled Lists	-0.12	-0.14	-0.11	1.67E+50
	Recalled Chunk Before	0.05	-0.05	0.14	0.12
	Recalled Singleton Before	0.05	-0.05	0.14	0.12

Table 3

Posteriors means, 95% HDIs, and the BFs of the parameters of the linear regression model fitted to the data of Experiment 2 in panels a and b in Figure 7, respectively.

		Measure			
		Posterior Mean	95% HDI		Bayes Factor
Chunk Last	**Effect**		Lower Bound	Upper Bound	
All Trials	Chunk (vs. Baseline)	0.02	-0.01	0.05	0.06
	Singleton (vs. Baseline)	0.24	0.21	0.27	2.60E+50
	Row	0.03	0.00	0.05	0.25
	Chunk x Row	0.05	-0.01	0.11	0.22
	Singleton x Row	-0.06	-0.12	0.00	0.36
	Singleton vs. Chunk	0.22	0.19	0.25	2.4e+413
	Nr. Previous Recallled Lists	-0.07	-0.09	-0.06	3.60E+18
	Recalled Chunk Before	-0.03	-0.11	0.04	0.09
	Recalled Singleton Before	0.03	-0.04	0.10	0.08
Chunks and Singletons Correct	Chunk	0.02	-0.02	0.06	0.05
	Singleton	0.24	0.20	0.28	1.70E+34
	Row	0.03	-0.03	0.08	0.08
	Chunk x Row	0.05	0.00	0.11	0.25
	Singleton x Row	-0.06	-0.12	-0.01	0.47
	Singleton - Chunk	0.22	0.17	0.27	1.30E+15
	Nr. Previous Recallled Lists	-0.07	-0.09	-0.06	6.40E+13
	Recalled Chunk Before	-0.02	-0.10	0.07	0.08
	Recalled Singleton Before	0.01	-0.07	0.10	0.08

Table 4

Posteriors means, 95% HDIs, and the BFs of the parameters of the linear regression model fitted to the data of Experiment 3 in panels a and b of Figure 10, respectively.

		Measure			
		Posterior Mean	95% HDI		Bayes Factor
Chunk First	**Effect**		Lower Bound	Upper Bound	
All Trials	Chunk (vs. Baseline)	0.04	0.01	0.08	0.61
	Singleton (vs. Baseline)	0.13	0.09	0.16	2.20E+09
	Row	0.07	0.00	0.15	0.38
	Chunk x Row	-0.02	-0.09	0.04	0.07
	Singleton x Row	-0.12	-0.24	-0.01	0.85
	Singleton vs. Chunk	0.08	0.05	0.12	116.00
	Nr. Previous Recallled Lists	-0.12	-0.15	-0.09	4.50E+11
	Recalled Chunk Before	-0.05	-0.13	0.03	0.15
	Recalled Singleton Before	0.08	-0.01	0.16	0.34
Chunks and Singletons Correct	Chunk (vs. Baseline)	0.06	0.02	0.12	2.20
	Singleton (vs. Baseline)	0.14	0.10	0.19	7.30E+07
	Row	0.06	0.03	0.14	15.00
	Chunk x Row	-0.02	-0.10	0.07	0.08
	Singleton x Row	-0.16	-0.24	-0.03	62.00
	Singleton - Chunk	0.08	0.04	0.14	13.00
	Nr. Previous Recallled Lists	-0.11	-0.13	-0.08	1.50E+22
	Recalled Chunk Before	-0.02	-0.13	0.09	0.09
	Recalled Singleton Before	0.09	-0.01	0.20	0.42

Table 5

Posteriors means, 95% HDIs, and the BFs of the parameters of the linear regression model fitted to the data of Experiment 3 in panels a and b of Figure 11, respectively.

		Measure			
			95% HDI		
Chunk Last	**Effect**	Posterior Mean	Lower Bound	Upper Bound	Bayes Factor
All Trials	Chunk (vs. Baseline)	0.03	-0.01	0.06	0.07
	Singleton (vs. Baseline)	0.16	0.12	0.19	1.30E+15
	Row	0.03	-0.09	0.14	0.10
	Chunk x Row	-0.01	-0.07	0.04	0.05
	Singleton x Row	-0.03	-0.09	0.03	0.08
	Singleton vs. Chunk	0.13	0.09	0.18	1’098’477.00
	Nr. Previous Recallled Lists	-0.08	-0.10	-0.06	1.60E+08
	Recalled Chunk Before	0.00	-0.11	0.10	0.09
	Recalled Singleton Before	-0.03	-0.12	0.06	0.09
Chunks and Singletons Correct	Chunk (vs. Baseline)	0.03	-0.02	0.08	0.09
	Singleton (vs. Baseline)	0.17	0.13	0.21	3.80E+11
	Row	0.01	-0.10	0.13	0.10
	Chunk x Row	-0.03	-0.11	0.04	0.09
	Singleton x Row	-0.04	-0.11	0.03	0.12
	Singleton - Chunk	0.14	0.08	0.19	1.00E+04
	Nr. Previous Recallled Lists	-0.08	-0.10	-0.06	2.40E+10
	Recalled Chunk Before	0.01	-0.12	0.14	0.10
	Recalled Singleton Before	-0.04	-0.16	0.07	0.12

Table 6

Posterior means, 95% HDIs, and the BFs of the parameters of the linear regression fitted to the data of Experiment 4 shown in panels a and b of Figure 14, respectively.

Chunk Before	Effect	Measure			
		Posterior Mean	95% HDI		Bayes Factor
			Lower Bound	Upper Bound	
All Trials	Chunk in Upper Row	0.07	0.04	0.10	825.00
	Chunk in Middle Row	0.05	0.01	0.08	1.20
	Row	0.04	-0.01	0.09	0.12
	Row x Chunk in Upper Row	-0.05	-0.10	0.00	0.37
	Nr. Previous Recallled Lists	-0.17	-0.20	-0.15	3.60E+42
	Recalled Chunk Before	-0.05	-0.10	0.00	0.24
Chunks Correct	Chunk in Upper-Row	0.08	0.04	0.11	878.00
	Chunk in Middle Row	0.07	0.03	0.11	9.20
	Row	0.03	-0.02	0.09	0.09
	Row x Chunk in Upper Row	-0.07	-0.13	-0.02	2.00
	Nr. Previous Recallled Lists	-0.18	-0.20	-0.15	7.90E+43
	Recalled Chunk Before	-0.06	-0.12	0.00	0.41

Table 7

Posterior means, 95% HDIs, and the BFs of the parameters of the linear regression fitted to the data of Experiment 4 shown in panels a and b of Figure 15, respectively.

Chunk After	Effect	Measure			
		Posterior Mean	95% HDI		Bayes Factor
			Lower Bound	Upper Bound	
All Trials	Chunk in Middle Row	0.07	0.04	0.11	125.00
	Chunk in Lower Row	0.00	-0.02	0.03	0.02
	Row	0.04	0.01	0.06	1.30
	Row x Chunk in Position 3	-0.01	-0.06	0.04	0.04
	Nr. Previous Recallled Lists	-0.09	-0.11	-0.08	4.60E+38
	Recalled Chunk Before	0.01	-0.04	0.05	0.04
Chunks Correct	Chunk in Middle Row	0.09	0.05	0.13	993.00
	Chunk in Lower Row	-0.01	-0.04	0.02	0.03
	Row	0.04	0.01	0.07	1.30
	Row x Chunk in Lower Row	0.01	-0.05	0.06	0.05
	Nr. Previous Recallled Lists	-0.09	-0.11	-0.07	2.50E+25
	Recalled Chunk Before	0.02	-0.03	0.08	0.06

Figure Captions

Figure 1. Serial recall accuracy (proportion correct) of new lists in Experiment 1. Recall of new lists with 2 and 4 items are presented in different sub-panels. Performance is plotted as a function of the size of the other list (2 or 4 items) for the two types of other lists (new list or chunk). Error bars represent within-subjects standard errors.

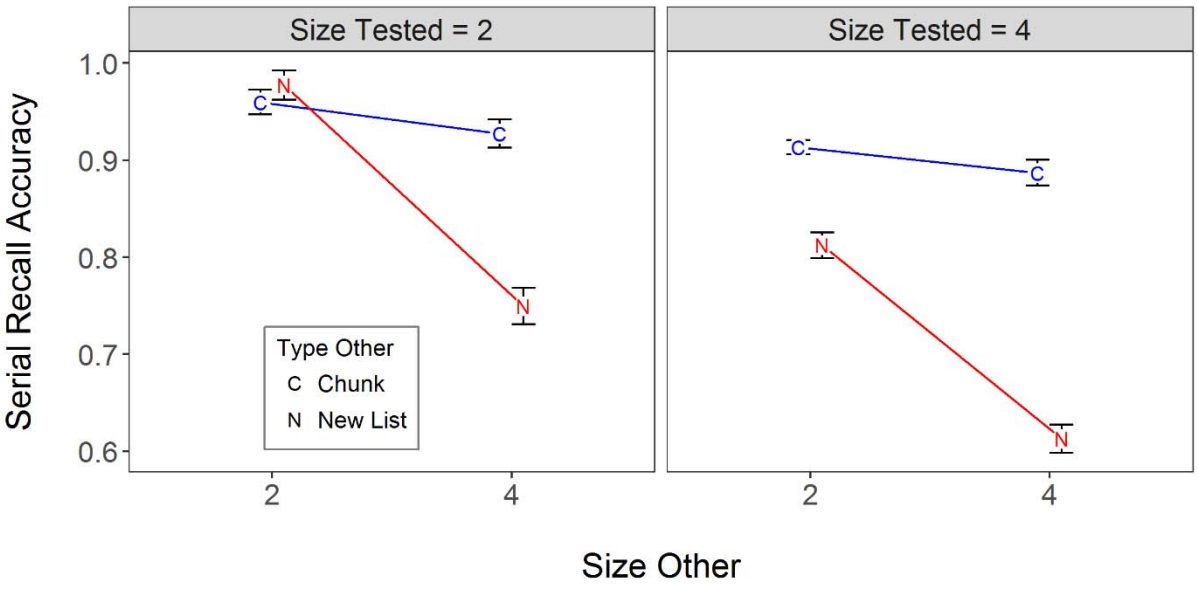

Figure 2. Serial recall accuracy (proportion correct) in all 16 conditions of Experiment 1.

Note. For better interpretability we omitted error bars.

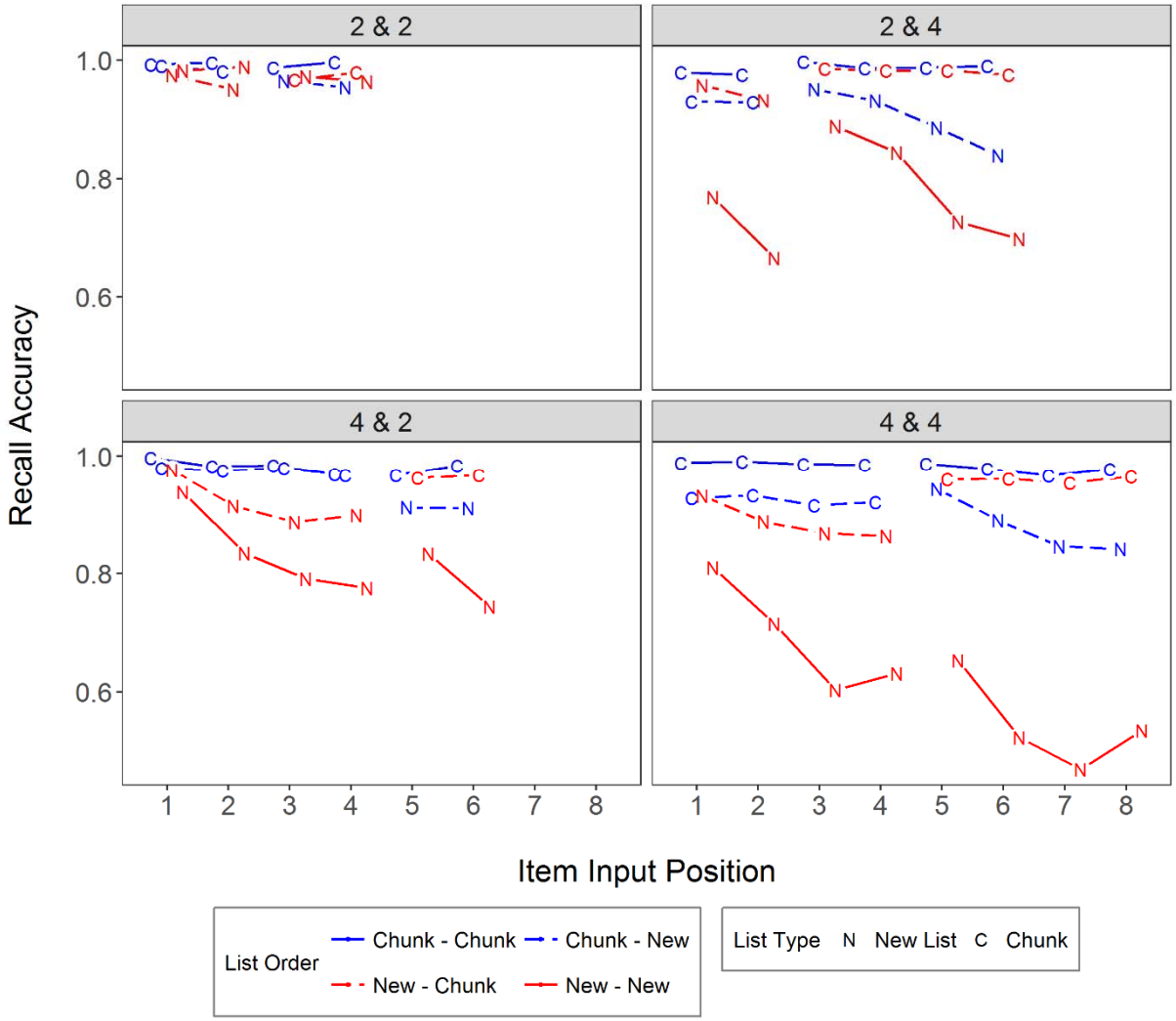

Figure 3. Example of the five experimental conditions in Experiment 2. Singleton lists (single letter) and chunked lists are identified in red (note that in the actual experiment all frames were black). The black dashed lines indicate frames in which no item was presented.

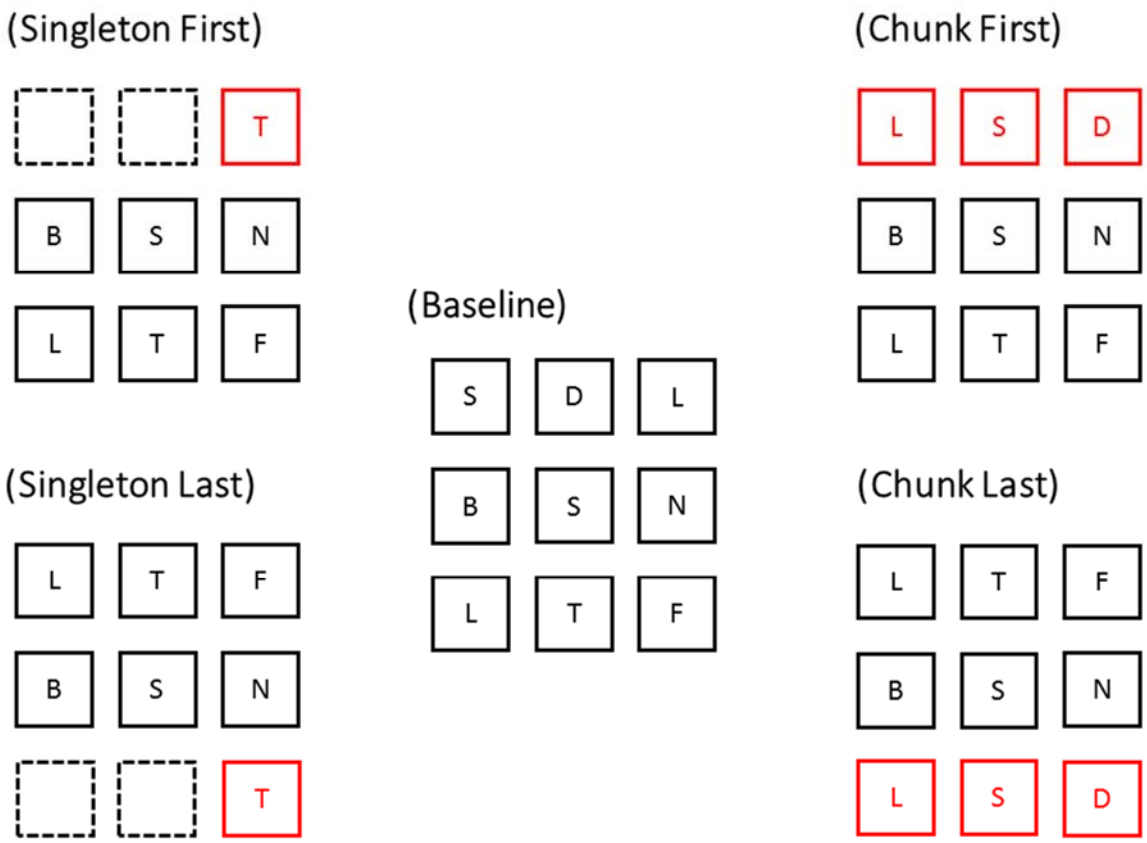

Figure 4. Serial recall accuracy (proportion correct) is plotted against item input position separately for each of the five experimental conditions in Experiment 2. For better interpretability of the figure, (a) we allocated singletons presented in the upper and lower row in the figure to item input position 3 and 7, respectively, and (b) we omitted error bars. U stands for Upper Row, L for Lower Row.

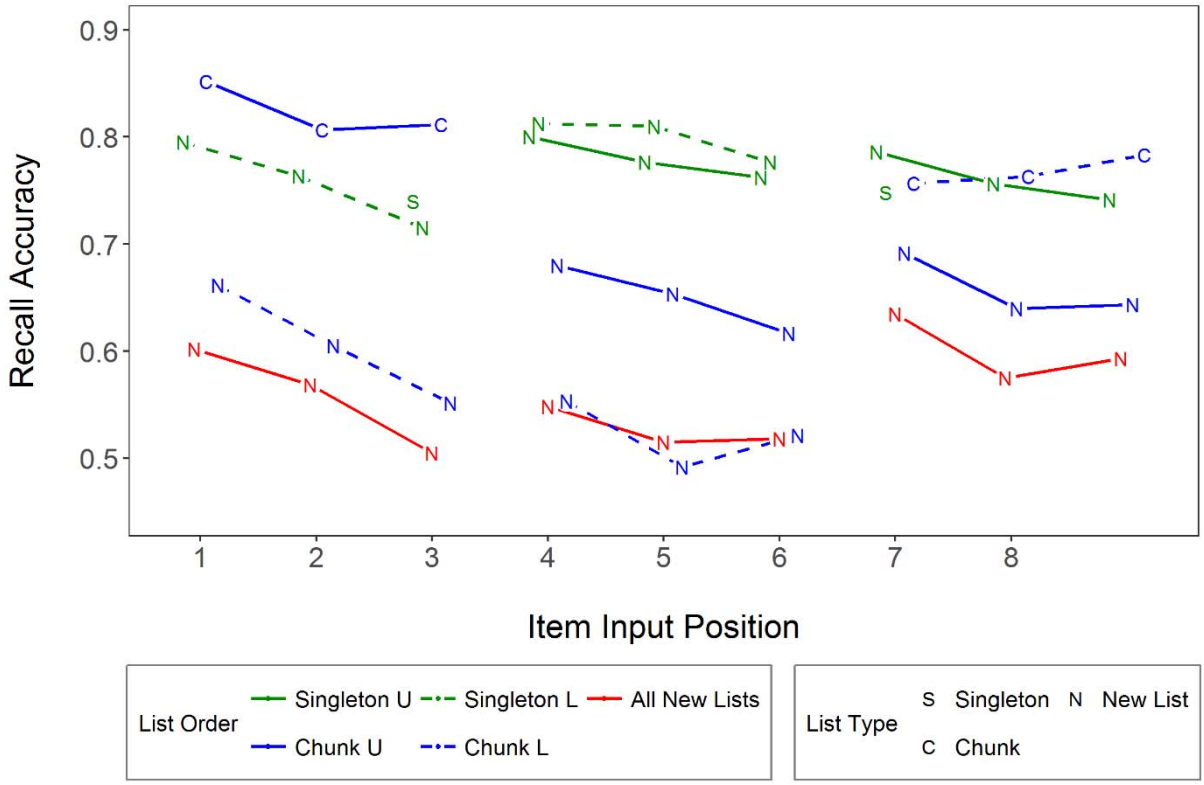

Figure 5. Serial recall accuracy of the three list types is plotted against row of presentation. The three panels represent the data from the three list output positions in Experiment 2.

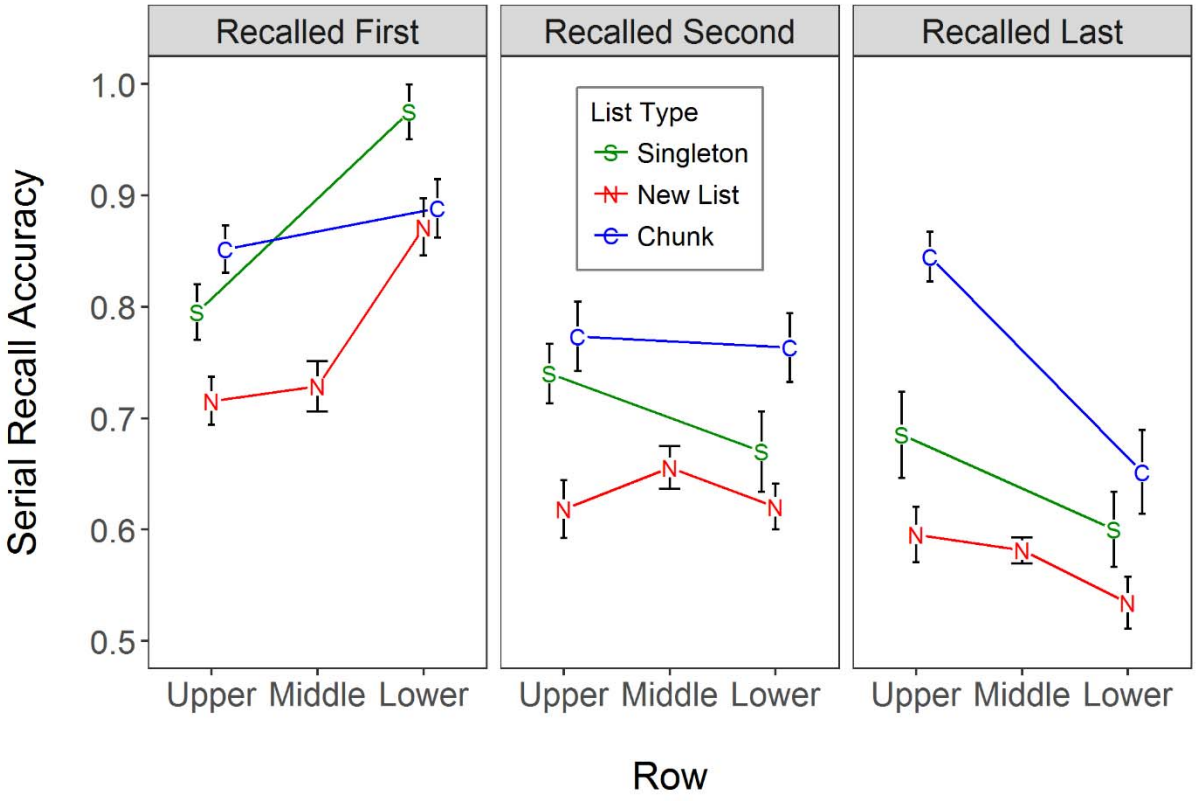

Figure 6. Mean serial recall accuracy of new lists that were preceded by a singleton, a new list, or a chunk plotted against row of presentation in Experiment 2. Panel a is based on all data, and Panel b is based only on those data from the Chunk First condition and the Singleton First condition in which participants recalled the chunk or the singleton correctly. The error bars represent within-subjects standard errors. U stands for Upper Row.

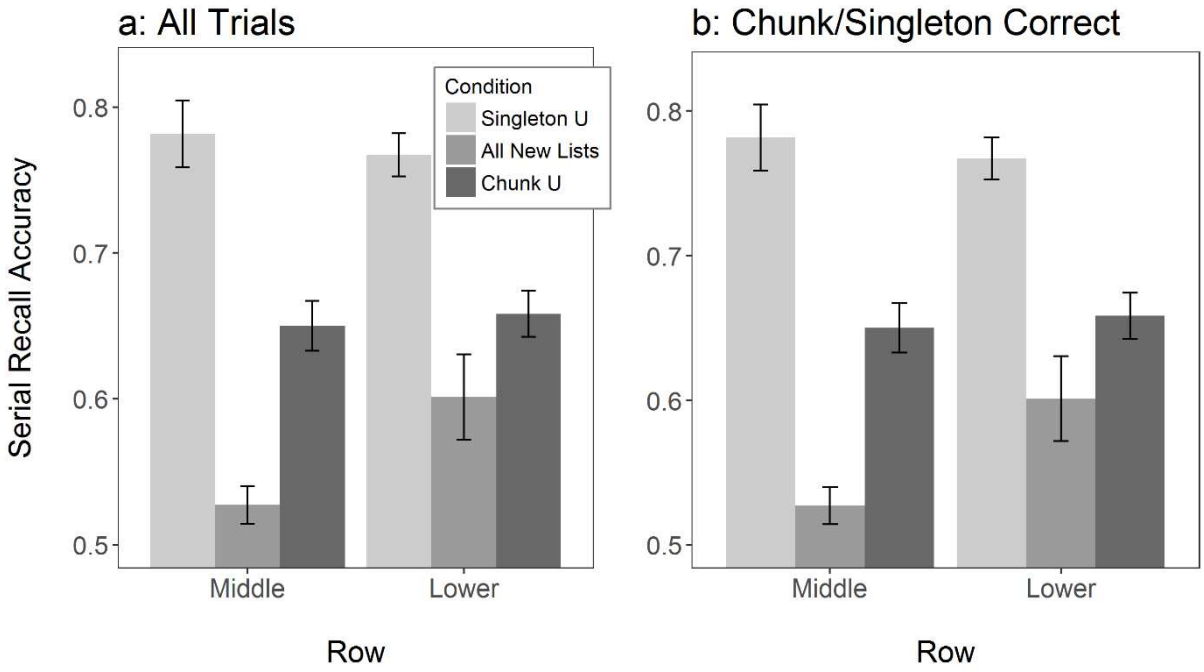

Figure 7. Mean serial recall accuracy of new lists when a singleton, a new list, or a chunk was presented in the lower row (Panel a: full data; panel b: Data from trials with correct recall of chunks/singletons) in Experiment 2. Error bars represent within-subjects standard errors. L stands for Lower Row.

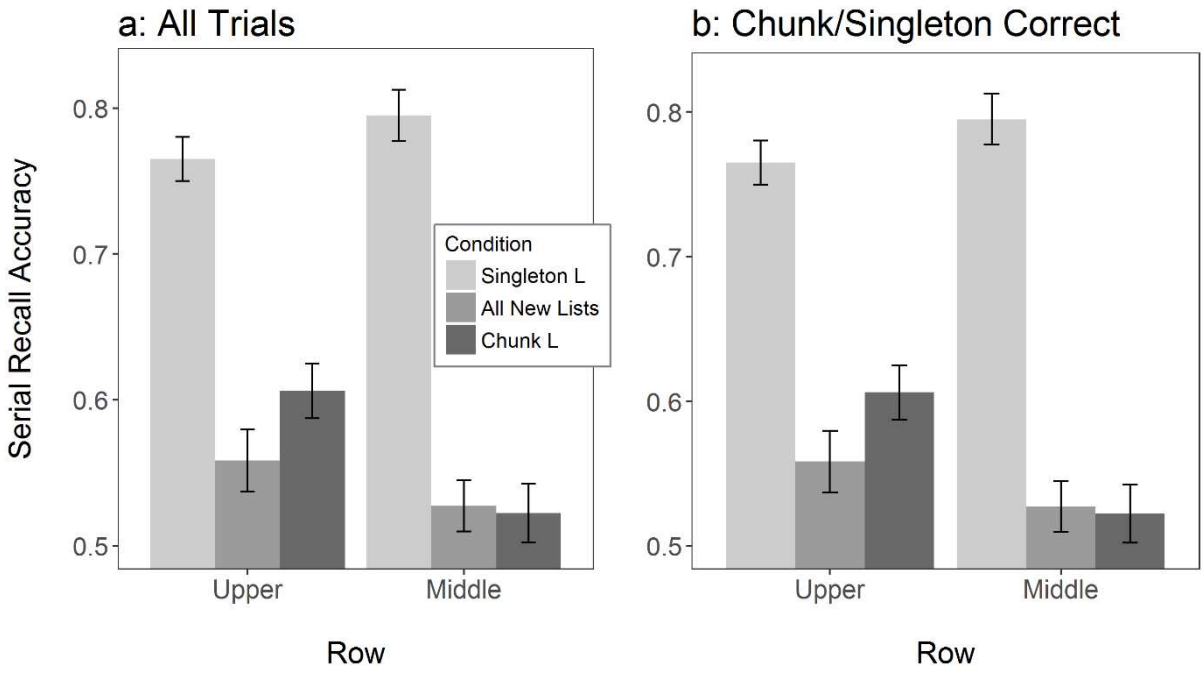

Figure 8. Serial recall accuracy (proportion correct) is plotted against item input position separately for each of the five experimental conditions in Experiment 3. For better interpretability of the figure, (a) we allocated singletons presented in the upper and lower row in the figure to item input position 3 and 7, respectively, and (b) we omitted error bars. U stands for Upper Row, L for Lower Row.

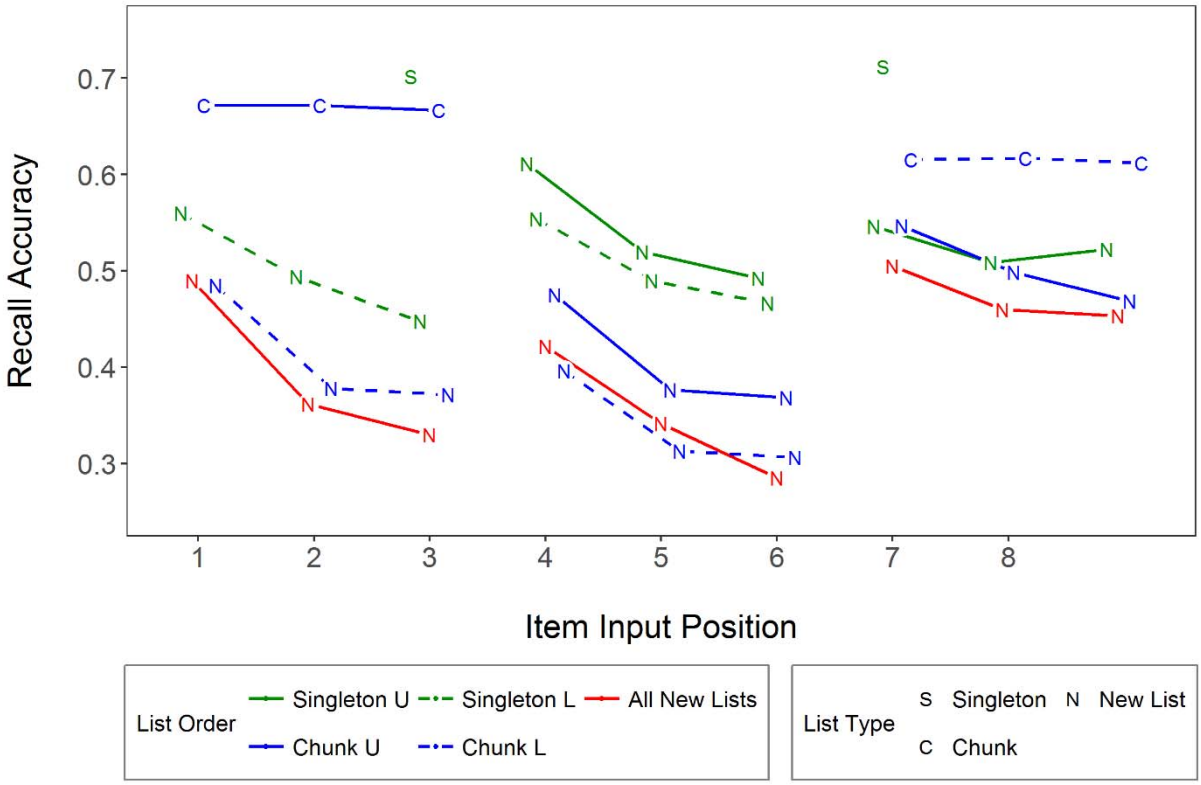

Figure 9. Serial recall accuracy (proportion correct) in Experiment 3 for the three list types plotted against row of presentation. Error bars represent within-subjects standard errors.

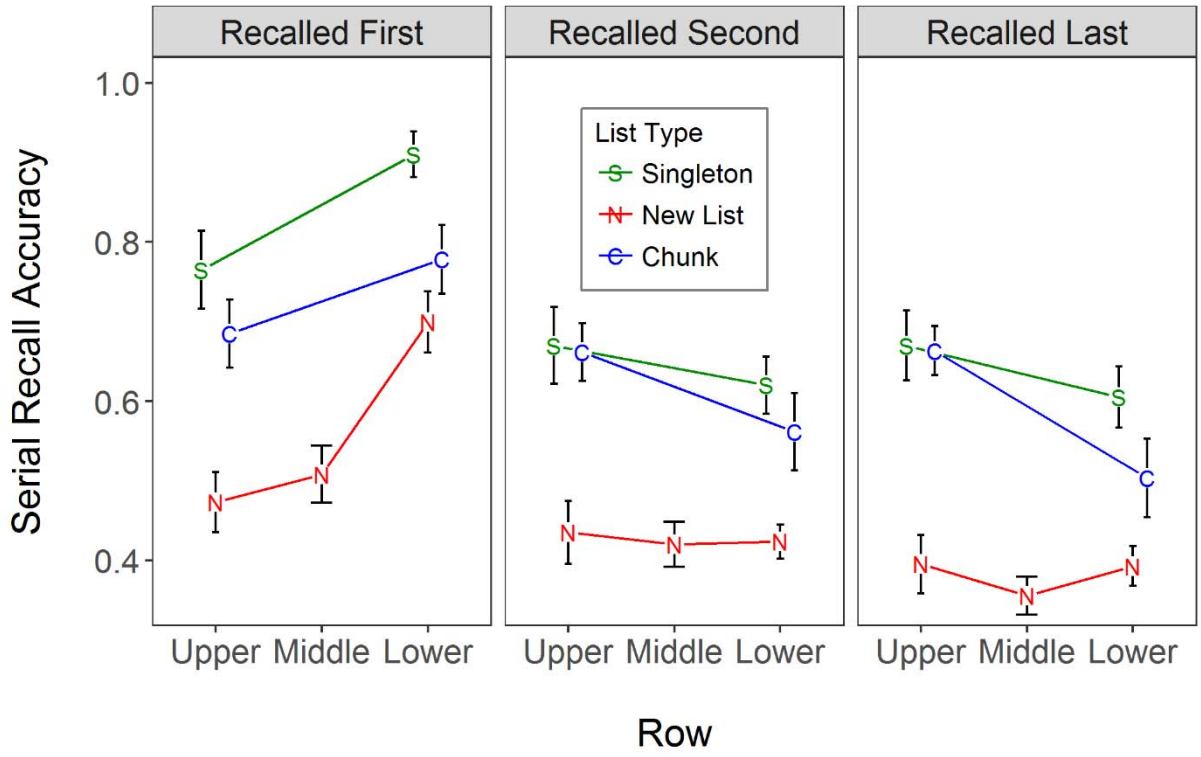

Figure 10. Mean serial recall accuracy of new lists that were preceded by a singleton, a new list, or a chunk plotted against row of presentation in Experiment 3. Panel a is based on all data, and Panel b is based only on those data from the Chunk First condition and the Singleton First condition in which participants recalled the chunk or the singleton correctly. The error bars represent within-subjects standard errors. U stands for Upper Row.

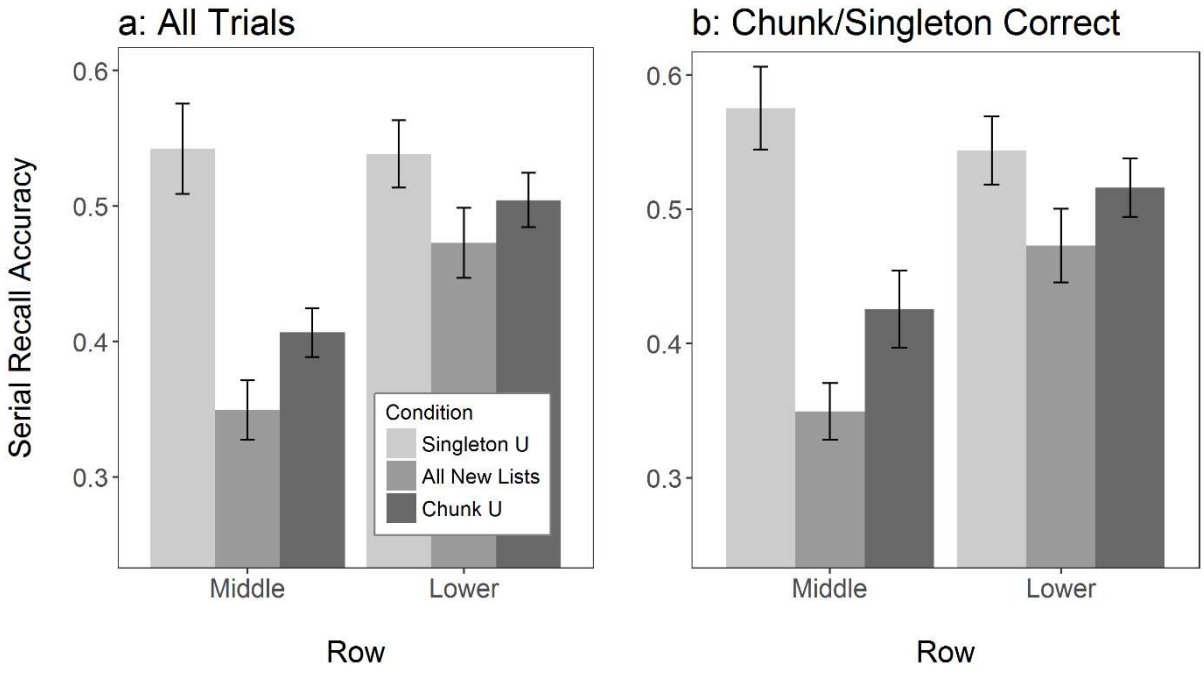

Figure 11. Mean serial recall accuracy of new lists when a singleton, a new list, or a chunk was presented in the lower row (Panel a: full data; panel b: Data from trials with correct recall of chunks/singletons) in Experiment 3. Error bars represent within-subjects standard errors. L stands for Lower Row.

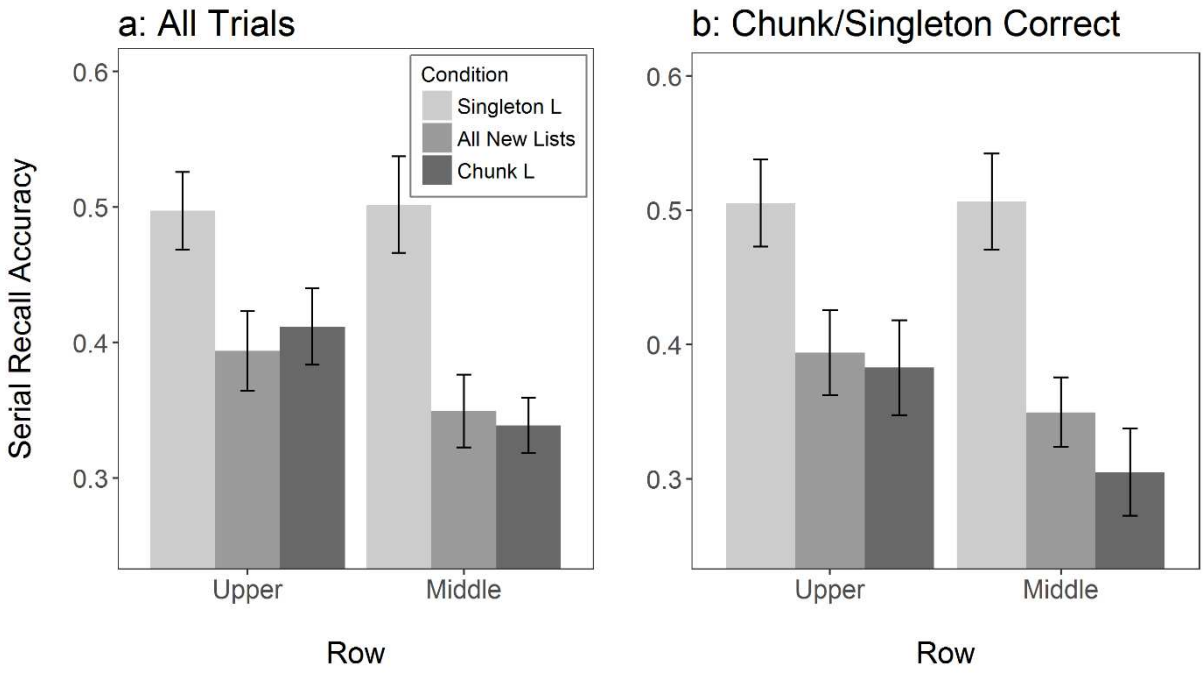

Figure 12. Serial recall accuracy (proportion correct) is plotted against item input position separately for each of the four experimental conditions in Experiment 4. For better interpretability of the figure we omitted error bars. U stands for Upper Row, M for Middle Row, and L for Lower Row.

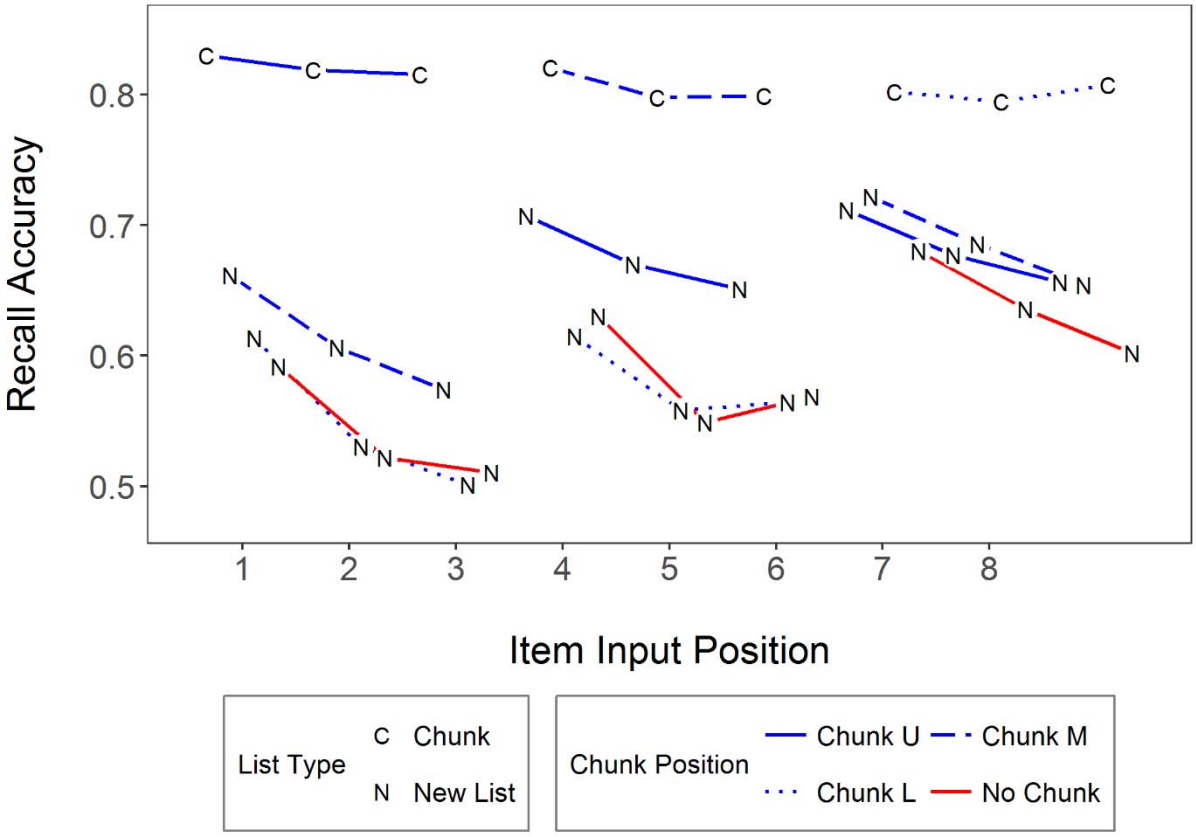

Figure 13. Serial recall accuracy in Experiment 4 for chunks and new lists plotted against row of presentation. The three panels represent the three list output positions.

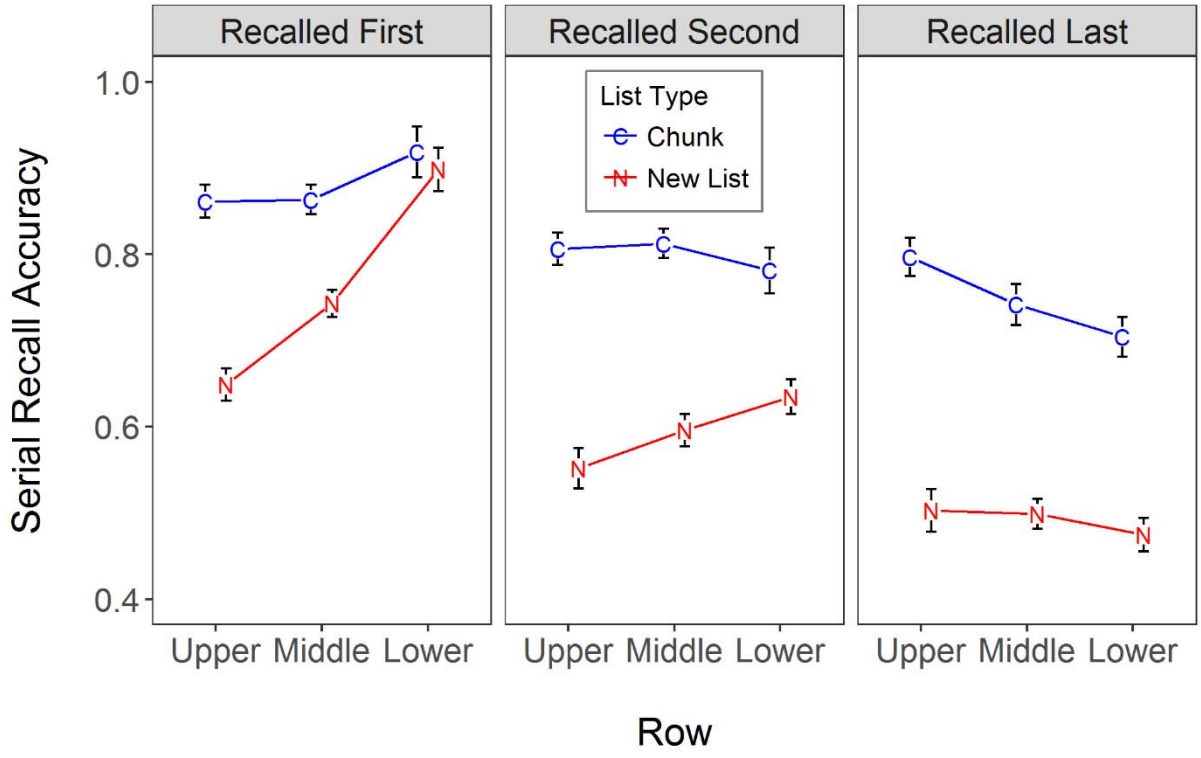

Figure 14. Serial recall accuracy of new lists that were preceded by a chunk compared to new lists in the Baseline condition of Experiment 4. Panel a shows data of all trials, whereas panel b subsets the data of trials with chunks in which chunks were recalled correctly. U stands for Upper Row, M for Middle Row.

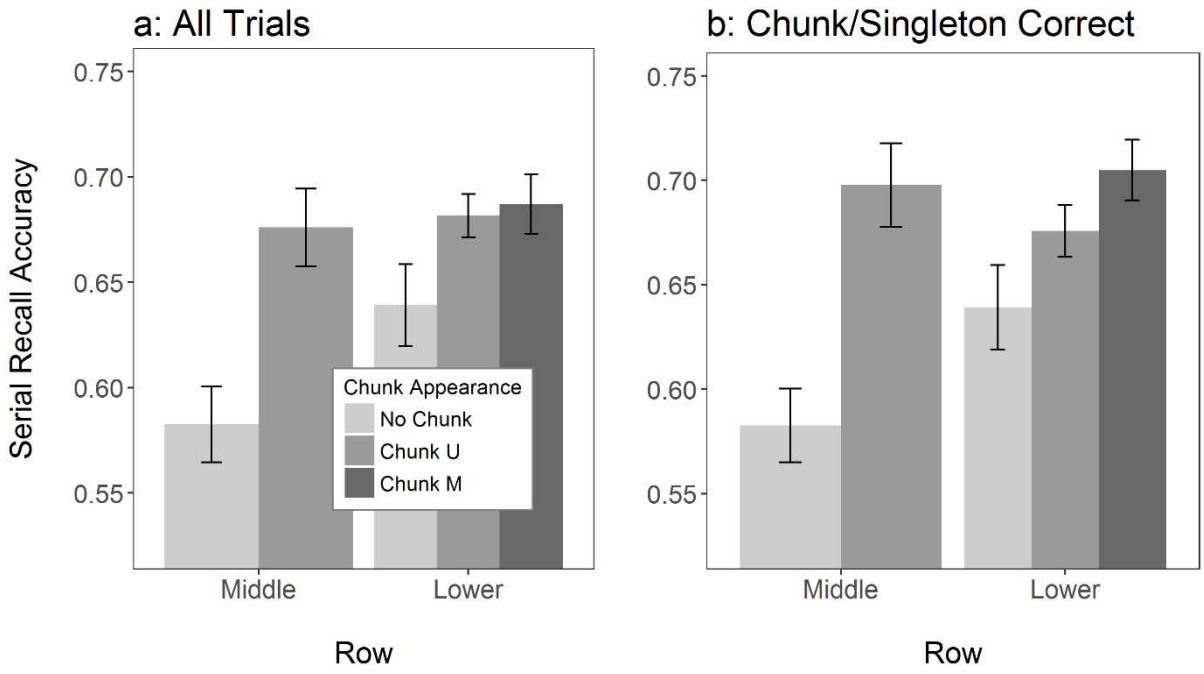

Figure 15. Serial recall accuracy of new lists that were followed by a chunk in Experiment 4. Panel a shows data of all trials, whereas panel b shows data of trials with chunks conditioned on correct recall of the chunk. M stands for Middle Row, L for Lower Row.